UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Renato Markele Ferreira Cândido

Reachability Analysis of Uncertain Max Plus Linear Systems

Análise de Alcançabilidade em Sistemas Max Plus Incertos

Campinas

2017

Renato Markele Ferreira Cândido

Reachability Analysis of Uncertain Max Plus Linear Systems

Análise de Alcançabilidade em Sistemas Max Plus Incertos

Thesis presented to the School of Electrical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Automation under the Cotutelle Agreement between Unicamp and the University of Angers (France).

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Automação no âmbito do Acordo de Cotutela firmado entre a Unicamp e a Universidade de Angers (França)

Supervisors/Orientadores: Prof. Dr. Rafael Santos Mendes, Prof. Dr. Laurent Hardouin Co-Supervisor/Coorientador: Prof. Dr. Mehdi Lhommeau

Este exemplar corresponde à versão final da tese defendida pelo aluno Renato Markele Ferreira Cândido, e orientada por Prof. Dr. Rafael Santos Mendes e Prof. Dr. Laurent Hardouin.

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): CNPq, 164765/2013-1; CAPES, 99999.002340/2015-01

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

C161r	Cândido, Renato Markele Ferreira, 1988- Reachability analysis of uncertain max plus linear systems / Renato Markele Ferreira Cândido. – Campinas, SP : [s.n.], 2017. Orientadores: Rafael Santos Mendes e Laurent Hardouin. Coorientador: Mehdi Lhommeau. Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. Em cotutela com: Université d'Angers. 1. Sistema a eventos discretos. 2. Sistemas incertos. 3. Verificação formal. I. Mendes, Rafael Santos,1957-. II. Hardouin, Laurent. III. Lhommeau, Mehdi. IV. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. VI. Título.

Informações para Biblioteca Digital

Título em outro idioma: Análise de alcançabilidade em sistemas max plus incertos
Palavras-chave em inglês:
Discrete event systems
Uncertain systems
Formal verification
Área de concentração: Automação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Rafael Santos Mendes [Orientador]
José Eduardo Ribeiro Cury
Laurent Hardouin
Mehdi Lhommeau
Isabel Demongodin
Paulo Augusto Valente Ferreira
Data de defesa: 23-06-2017
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Renato Markele Ferreira Cândido RA: 123228

Data da Defesa: 23 de junho de 2017

Título da Tese: “Reachability Analysis of Uncertain Max Plus Linear Systems” / "Análise de Alcançabilidade em Sistemas Max Plus Incertos”

Prof. Dr. Rafael Santos Mendes (Presidente, FEEC/UNICAMP)
Prof. Dr. José Eduardo Ribeiro Cury (UFSC)
Prof. Dr. Laurent Hardouin (Université d’Angers)
Prof. Dr. Mehdi Lhommeau (Université d’Angers)
Profa. Dra. Isabel Demongodin (Aix-Marseille Université)
Prof. Dr. Paulo Augusto Valente Ferreira (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

To my little nephew Guilherme, to whom I wish wisdom and strength to face the difficulties and reach *his goals.*

To the people of Nelson de Sena, MG, Brasil. May this be the first of many PhD thesis in town.

Acknowledgements

I thank you God for this amazing journey of personal and professional growth. For putting amazing people in my path, whom I also owe thanks to.

I thank my supervisors Rafael, Laurent and Mehdi for giving me the precious opportunity to go to France to pursue my PhD, for the efforts they have made to solve all bureaucratic issues, for the priceless reception in France, for the valuable comments and discussions that helped me develop my research and, of course, for the great dinners we had, the French cuisine is really wonderful.

I would like to thank to the people from LARIS-ISTIA, where I had a good environment to develop my research, and to the professors from FEEC-UNICAMP, for the great courses offered.

I thank to the Brazilian funding agencies CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*) and CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*), for the financial support.

During my stay in Angers-France, I met very special people. I thank to the couple Nicolás and Patricia, for all the happy times we spent together, to everyone at Gym Pulse/Angers, where I had good times. In special, I owe a lot of thanks to my friend and ex-housemate Javier, who always helped me when I was in need and who was a great traveling companion.

My stay in Campinas would not be so happy if I had not met my housemates. I would like to thank them for the support, laughter, joking and for the tasty weekend lunches.

I owe a lot of thanks to my beloved parents Geraldo and Maria da Luz, whose immeasurable effort allowed me to get here. Thanks to my sisters Cristiane and Camila, to my uncles Corina and José Márcio, to my cousin Paulo César, to my brother-in-law Willian and to my girlfriend Bianca, for all the support and love that you all gave me on this long journey.

Difficult to see. Always in motion is the future.
(Yoda)

Abstract

Discrete Event Dynamic Systems (DEDS) are discrete-state systems whose dynamics are entirely driven by the occurrence of asynchronous events over time. Linear equations in the max-plus algebra can be used to describe DEDS subjected to synchronization and time delay phenomena. The reachability analysis concerns the computation of all states that can be reached by a dynamical system from an initial set of states. The reachability analysis problem of Max Plus Linear (MPL) systems has been properly solved by characterizing the MPL systems as a combination of Piece-Wise Affine (PWA) systems and then representing each component of the PWA system as Difference-Bound Matrices (DBM). The main contribution of this thesis is to present a similar procedure to solve the reachability analysis problem of MPL systems subjected to bounded noise, disturbances and/or modeling errors, called uncertain MPL (uMPL) systems. First, we present a procedure to partition the state space of an uMPL system into components that can be completely represented by DBM. Then we extend the reachability analysis of MPL systems to uMPL systems. Moreover, the results on reachability analysis of uMPL systems are used to solve the *conditional reachability problem*, which is closely related to the support calculation of the probability density function involved in the stochastic filtering problem.

Keywords: Reachability Analysis; Conditional Reachability Analysis; Max Plus Linear Systems; Piece-Wise Affine Systems; Difference-Bound Matrices.

Resumo

Os Sistemas a Eventos Discretos (SEDs) constituem uma classe de sistemas caracterizada por apresentar espaço de estados discreto e dinâmica dirigida única e exclusivamente pela ocorrência de eventos. SEDs sujeitos aos problemas de sincronização e de temporização podem ser descritos em termos de equações lineares usando a álgebra max-plus. A análise de alcançabilidade visa o cálculo do conjunto de todos os estados que podem ser alcançados a partir de um conjunto de estados iniciais através do modelo do sistema. A análise de alcançabilidade de sistemas Max Plus Lineares (MPL) pode ser tratada por meio da decomposição do sistema MPL em sistemas PWA (*Piece-Wise Affine*) e de sua correspondente representação por DBM (*Difference-Bound Matrices*). A principal contribuição desta tese é a proposta de uma metodologia similar para resolver o problema de análise de alcançabilidade em sistemas MPL sujeitos a ruídos limitados, chamados de sistemas MPL incertos ou sistemas uMPL (*uncertain Max Plus Linear Systems*). Primeiramente, apresentamos uma metodologia para particionar o espaço de estados de um sistema uMPL em componentes que podem ser completamente representados por DBM. Em seguida, estendemos a análise de alcançabilidade de sistemas MPL para sistemas uMPL. Além disso, a metodologia desenvolvida é usada para resolver o problema de análise de alcançabilidade condicional, o qual esta estritamente relacionado ao cálculo do suporte da função de probabilidade de densidade envolvida no problema de filtragem estocástica.

Palavras-chaves: Análise de Alcançabilidade; Análise de Alcançabilidade Condicional; Sistemas Max Plus Lineares; Sistemas PWA; DBM.

List of Figures

List of Tables

Contents

1 Introduction

Discrete Event Dynamic Systems (DEDS) are discrete-state systems whose dynamics are entirely driven by the occurrence of asynchronous events over discrete time instants (CASSANDRAS; LAFORTUNE, 2009, Sec. 1.3.2). Examples of DEDS include computer systems, telecommunication networks, manufacturing lines and transportation systems. The dynamics of such systems is often subjected to conflict, synchronization and time delay phenomena. In a manufacturing line, for instance, a conflict appears when two or more parts needs to be processed in a machine, at the same time, and it is necessary to decide which part will be processed first. Synchronization requires the availability of several parts at the same time. In a railway station, synchronization appears when a departing train must wait for certain incoming trains in order to allow changeover of passengers. Time delay can be associated to processing or traveling times, for instance.

DEDS subjected only to synchronization and time delay phenomena can be described in terms of linear equations using the Max-Plus Algebra. The max-plus algebra is an idempotent semiring, an algebraic structure also called dioid (BACCELLI *et al.*, 1992), in which the operations of sum ($\oplus$) and product ($\otimes$) are defined as the maximization and addition, respectively. Synchronization phenomena are modeled thanks to maximization: the start of a task waits for the completion of the preceding tasks, while the delay phenomena are modeled thanks to the classical sum: the completion time of a task is equal to the starting time plus the task duration. Consider a railway station in which a departing train must wait for all incoming trains. Assuming that the trains leave as soon as possible, the departure time of a train is given by the maximum of the arrival times of all incoming trains. The arrival time at a station is the sum of the departure time from the previous station plus the traveling time, assumed to be known.

The linearity property has advantaged the emergence of a specific theory for the performance analysis (HEIDERGOTT *et al.*, 2006) and the control of these systems, e.g., optimal open loop control (COHEN *et al.*, 1999; LHOMMEAU *et al.*, 2005) and optimal state-feedback control. Among closed-loop strategies we can cite the model matching problem (LHOMMEAU *et al.*, 2003) and the control strategies allowing the state to stay in a specific state subspace or semimodule (AMARI *et al.*, 2012; KATZ, 2007; MAIA *et al.*, 2011; NECOARA *et al.*, 2009; GONÇALVES *et al.*, 2016).

The entries of Max-Plus Linear (MPL) system matrices are associated to system delays such as processing or traveling times. These parameters are often subjected to noise and disturbances, which should be taken into account in order to avoid tracking error or closed loop instability (van den Boom; De Schutter, 2002). In general, these perturbations are max-plus-multiplicative and appear as uncertainties in the max-plus model parameters. As a result the system matrices are uncertain. The Stochastic Max-Plus Linear (SMPL) systems are defined as MPL systems where the matrices entries are characterized by random variables (OLSDER *et al.*, 1990; HEIDERGOTT, 2006; van den Boom; De Schutter, 2002; DILORETO *et al.*, 2010; HARDOUIN *et al.*, 2010).

To assess whether the system reaches a certain state from a set of initial conditions is of great interest in many applications and concerns the reachability analysis. Consider for instance the safety analysis problem (MITCHELL, 2007): given a system and a set of initial states, the safety analysis aims to determine if the system can enter a specified set of unsafe states. The reachability analysis can be used to determine whether trajectories of the given system can reach the unsafe set from the initial set. Gazarik *et al.* (1999) use residuation to determine if a state is reachable, via an MPL model, from a single initial condition and to generate a control sequence to reach it. Gaubert e Katz (2003), show that if the initial set is a rational semimodule the reachable set is also a rational semimodule. These authors mention that this set has a "simple shape" and suggest that an efficient numerical method remains to be designed. In Lu *et al.* (2012) reachability analysis of timed automata is tackled by considering max-plus polyhedra, a more general class of sets than semimodules. For a more exhaustive presentation on max-plus polyhedra, see Allamigeon *et al.* (2008). However, it is not possible to employ related techniques for reachability analysis of MPL systems since the two modeling frameworks are not comparable.

Under the requirement that the set of initial states is a max-plus polyhedron, forward reachability analysis can be performed over max-plus algebra. Similarly, under the same requirements, backward reachability analysis can be performed over the max-plus algebra, where in addition the system matrix has to be max-plus invertible. Computationally, the approach based on max-plus polyhedra can be advantageous since its time complexity is polynomial. However, the requirements limit the applicability of the approach. To the best of the author's knowledge there exist no general approach for reachability analysis over max-plus algebra. In Adzkiya *et al.* (2014b), forward reachability analysis of autonomous MPL systems is alternatively addressed by characterizing the MPL system as a Piece-Wise Affine (PWA) system and then representing the PWA system as a collection of Difference Bound Matrices (DBM) (DILL, 1990). It is shown that, if the initial set is depicted as the union of finitely many DBM, then the set of all states that can be reached via the model dynamics, at any given event step, can also be depicted as the union of finitely many DBM, and therefore it is possible to map DBM-sets through MPL systems. The authors state that any max-plus polyhedra can be depicted as a union of DBM and claim that their approach is more general than the one using max-plus polyhedra, the price to pay being a potential explosion in the number of DBM during computations. Moreover in Adzkiya *et al.* (2014a), the approach has also been applied to backward reachability analysis of autonomous MPL systems considering a final set depicted as union of DBM despite the non invertibility of the max-plus linear system. In Adzkiya *et al.* (2015), these results have been extended to nonautonomous MPL systems. Experiments carried out in Adzkiya *et al.* (2015, Sec. 5) suggest that the potential explosion in the number of DBM is not a problem and allows claiming the applicability of the approach.

To describe an MPL system by means of DBM it is necessary to express it as a Piece-Wise Affine System (PWA). This is always possible (HEEMELS *et al.*, 2001) and it is done by partitioning the state space into regions in which the system can be modeled by affine equations (in classical algebra). The PWA system is the union of these affine subsystems and the key point is that each affine system and its corresponding active state space region can be independently represented by one DBM (see section 2.5.1). The main advantage of this representation is the existence of many efficient algorithms for DBM manipulation and its drawback is the upsizing of the representation of a MPL system from one compact state equation to multiple DBM.

In this work, we aim to use a similar approach to analyze systems where the uncertain parameters can vary over a known interval, herein defined as uncertain MPL (uMPL) systems, as detailed in chapter 4. We do not seek to provide any stochastic analysis of these systems. Thus, for the purposes of this work, the uMPL systems are treated as non-deterministic systems (rather than stochastic systems). The approach is synthesized as follows. First, we present a procedure to partition the uMPL systems into subsystems that can be fully represented by DBM. Then, we show that the image and the inverse image of a DBM w.r.t. each subsystem of the partitioned uMPL system is again a DBM. This result made it possible to extend most of the results presented in (ADZKIYA *et al.*, 2014b; ADZKIYA *et al.*, 2014a; ADZKIYA *et al.*, 2015) to uMPL systems. Then, for the forward reachability analysis, given a set of initial conditions represented by a union of finitely many DBM, we present a procedure to compute the sets of all states that *can be* reached at each event step, which can also be represented by a union of finitely many DBM. Similarly, for the backward reachability analysis, given a set of final conditions represented by a union of finitely many DBM, we present a procedure to compute the sets of all states that *may lead* to the set of final conditions in a given number of steps. We also present a residuation-based procedure to compute the inverse image of a point that is less expensive than the procedure based on the system partitioning.

Furthermore, we use the results on reachability analysis of uMPL system to solve the *conditional reachability problem*. The conditional reachability analysis concerns the computation of the set of all states that may be reached from a set of initial states, in a given event step, conditioned to a sequence of measures related to the state through an uMPL equation. Closely related to conditional reachability is the filtering problem. Bayesian methods provide a rigorous general framework for filtering problem (GORDON *et al.*, 1993). The objective of the Bayesian state estimation is to construct the posterior Probability Density Function (PDF) of the states based on all information available. In this context, the conditional reachability analysis corresponds to the support calculation of the posterior PDF of the uMPL system states. However, it should be noted that the conditional reachability problem is not stochastic since it does not lead to an estimate of any probabilistic measure. As an example of application, the conditional reachability analysis could be useful to improve Particle Filtering algorithms. Particle Filters, or Sequential Monte Carlo methods, are suboptimal Bayesian algorithms based on weighted-particle approximation of probability densities (ARULAMPALAM *et al.*, 2002; DOUCET *et al.*, 2000). Particle filters applied to Max-Plus systems have been studied in Silva *et al.* (2011), CÂNDIDO *et al.* (2013), CÂNDIDO e MENDES (2014).

This work is organized as follows: Chapter 2 recalls the MPL systems and their decompositions as PWA systems, as well as the DBM representation of PWA systems generated by MPL systems. Chapter 3 gives an overview of the methods for reachability analysis of MPL systems presented in Adzkiya *et al.* (2014b), Adzkiya *et al.* (2014a), Adzkiya *et al.* (2015). The main contribution appears in Chapter 4 which introduces the uMPL systems and their descriptions by means of DBM. Chapter 5 extends reachability analysis to uMPL systems. Chapter 6 defines and solve the conditional reachability problem by using the results on reachability analysis for uMPL. Finally, Chapter 7 concludes the work. We shall remark that chapters 4, 5 and 6 are based on a paper submitted to Automatica (Journal of IFAC), which is, currently, in the third round of review (CÂNDIDO *et al.*, 2017, Under Review to Automatica).

2 Preliminaries

2.1 Idempotent Semirings

This section recall some basic concepts of *idempotent semirings*, an algebraic structure also known as *dioids* (COHEN *et al.*, 1989; BACCELLI *et al.*, 1992).

Definition 2.1 (Idempotent semirings (COHEN *et al.*, 1989, Def. 1)) *A set S, endowed with two internal operations: $\oplus$ (sum) and $\otimes$ (product); is an* idempotent semiring *or* dioid *if the following axioms are verified:*

Axiom 2.1 (Associativity)

$$\forall a, b, c \in S \begin{cases} (a \oplus b) \oplus c = a \oplus (b \oplus c) \\ (a \otimes b) \otimes c = a \otimes (b \otimes c) \end{cases}$$

Axiom 2.2 (Commutativity of addition)

$$\forall a, b \in S \;\; a \oplus b = b \oplus a$$

Axiom 2.3 (Distributivity of multiplication w.r.t addition)

$$\forall a, b, c \in S \begin{cases} (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \\ c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b) \end{cases}$$

Axiom 2.4 (Existence of a zero element ε and an identity element e)

$$\exists \varepsilon \in S : \forall a \in S, \;\; a \oplus \varepsilon = a$$

$$\exists e \in S : \forall a \in S, \;\; a \otimes e = a$$

Axiom 2.5 (Absorbing zero element)

$$\forall a \in S, \;\; a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$$

Axiom 2.6 (Idempotency of addition)

$$\forall a \in S, \;\; a \oplus a = a$$

Table 1 – Idempotent Semirings

S	$\oplus$	$\otimes$	ε	e	Application	Notation
$\mathbb{R} \cup \{+\infty\}$	min	$+$	$+\infty$	0	shortest path	$\mathbb{R}_{min}$
$\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$	min	$+$	$+\infty$	0	shortest path	$\overline{\mathbb{R}}_{min}$
$\mathbb{R} \cup \{-\infty\}$	max	$+$	$-\infty$	0	widest path	$\mathbb{R}_{max}$
$\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$	max	$+$	$-\infty$	0	longest path	$\overline{\mathbb{R}}_{max}$
$\mathbb{R}^+ \cup \{-\infty\}$	max	min	0	$+\infty$	max capacity	$\mathbb{R}^+_{max,min}$
$[0,1]$	max	$\times$	0	1		
$\mathbb{R}^+$	max	$\times$	0	1		$\mathbb{R}^+_{max,\times}$
$\{0,1\}$	$\cup$	$\cap$	0	1	logic	$\mathbb{B}$

In Table 1, taken from (QUADRAT, 1999, Chap. 1), are some examples of idempotent semirings and its applications.

As in the classical algebra, the k^{th} power of $a \in S$, denoted by $a^{\otimes k}$, is defined as $a^{\otimes k} = a^{\otimes k-1} \otimes a$, with $a^{\otimes 0} = e$.

In a dioid S, one has the following equivalence (BACCELLI *et al.*, 1992, Th. 4.28):

$$\forall a, b \in S,\ a = a \oplus b \Longleftrightarrow \exists c \in S : a = b \oplus c. \tag{2.1}$$

This equivalence defines a partial order relation noted by $\succeq$ as follows:

$$a \succeq b \Longleftrightarrow a = a \oplus b. \tag{2.2}$$

This relation is compatible with sum and with left and right product, i.e.:

$$a \succeq b \Longrightarrow \begin{cases} a \oplus c \succeq b \oplus c, & \text{(sum)} \\ a \otimes c \succeq b \otimes c, & \text{(right product)} \\ c \otimes a \succeq c \otimes b, & \text{(left product)} \end{cases}$$

Definition 2.2 (Complete dioid (BACCELLI *et al.*, 1992, Def. 4.32)) *A dioid is complete if it is closed for infinite sums and Axiom 2.3 extends to infinite sums.*

In a complete dioid the top element, denoted $\top$, exists and it is equal to the sum of all elements in S (BACCELLI *et al.*, 1992, Sec. 4.3.3):

$$\top = \bigoplus_{x \in S} x. \tag{2.3}$$

This element is absorbing for addition since $\forall a$, $\top \oplus a = \top$. Besides, according to axiom 2.5 $\top \otimes \varepsilon = \varepsilon$.

For a complete dioid, a new inner operation representing the lower bound of the operands, denoted by $\wedge$, can be constructed (BACCELLI *et al.*, 1992, Sec. 4.3.4). The partial order relation presented in (2.2) can be expressed as:

$$a \succeq b \Longleftrightarrow a = a \oplus b \Longleftrightarrow b = a \wedge b. \tag{2.4}$$

This operation is associative, commutative, idempotent and has $\top$ as neutral element: $\forall a,\ \top \wedge a = a$. This operation has also a property called absorption law (DUBREIL; DUBREIL-JACOTIN, 1964, p. 184), given by:

$$\forall a, b \in S,\ a \wedge (a \oplus b) = a \oplus (a \wedge b) = a. \tag{2.5}$$

Moreover, $\otimes$ is "subdistributive" w.r.t. $\wedge$ (BACCELLI *et al.*, 1992, Sec. 4.3.4):

$$\forall a, b, c \in S, \begin{cases} c \otimes (a \wedge b) \leq (c \otimes a) \wedge (c \otimes b), \\ (a \wedge b) \otimes c \leq (a \otimes c) \wedge (b \otimes c). \end{cases} \tag{2.6}$$

Neither the operation $\wedge$ necessarily distribute over $\oplus$ or $\oplus$ necessarily distribute over $\wedge$. However, $\oplus$ is "subdistributive" with respect to $\wedge$, and $\wedge$ is "superdistributive" with respect to $\oplus$ (BACCELLI *et al.*, 1992, Sec. 4.3.5), (COHEN *et al.*, 1989, Sec. 2.2):

$$\forall a, b, c \in S, \begin{cases} (a \wedge b) \oplus c \leq (a \oplus c) \wedge (b \oplus c), \\ (a \oplus b) \wedge c \geq (a \wedge c) \oplus (b \wedge c). \end{cases} \tag{2.7}$$

Definition 2.3 (Distributive dioid (BACCELLI *et al.*, 1992, Def. 4.39)) *A dioid S is distributive if it is complete and, for all subsets C of S,*

$$\forall a \in S, \begin{cases} \left(\bigwedge_{c \in C} c\right) \oplus a = \bigwedge_{c \in C} (c \oplus a), \\ \left(\bigoplus_{c \in C} c\right) \wedge a = \bigoplus_{c \in C} (c \wedge a). \end{cases}$$

Note that, if S is distributive, the equality holds in (2.7).

The sum and product of matrices are defined as follows: If A, B and C are, respectively, $n \times p$, $n \times p$ and $p \times q$ matrices with entries in a dioid S, then:

$$\begin{aligned} (A \oplus B)_{ij} &= a_{ij} \oplus b_{ij}, \quad i \in \{1, ..., n\},\ j \in \{1, ..., p\} & (2.8) \\ (A \otimes C)_{ij} &= \bigoplus_{k=1}^{p} (a_{ik} \otimes b_{kj}), \quad i \in \{1, ..., n\},\ j \in \{1, ..., q\}. & (2.9) \end{aligned}$$

Example 2.4 *Consider the matrices A, B and C with entries in $\mathbb{R}_{max}$ (see Table 1), where:*

$$A = \begin{pmatrix} 2 & 3 & e \\ \varepsilon & e & 4 \\ 4 & 1 & \varepsilon \end{pmatrix}, \; B = \begin{pmatrix} 1 & \varepsilon & e \\ 3 & 4 & 2 \\ 3 & 1 & e \end{pmatrix} \; \textit{and } C = \begin{pmatrix} 1 \\ e \\ 2 \end{pmatrix}.$$

Then:

$$A \oplus B = \begin{pmatrix} 2 \oplus 1 & 3 \oplus \varepsilon & e \oplus e \\ \varepsilon \oplus 3 & e \oplus 4 & 4 \oplus 2 \\ 4 \oplus 3 & 1 \oplus 1 & \varepsilon \oplus e \end{pmatrix} = \begin{pmatrix} 2 & 3 & e \\ 3 & 4 & 4 \\ 4 & 1 & e \end{pmatrix},$$

$$A \otimes C = \begin{pmatrix} 2 \otimes 1 \oplus 3 \otimes e \oplus e \otimes 2 \\ \varepsilon \otimes 1 \oplus e \otimes e \oplus 4 \otimes 2 \\ 4 \otimes 1 \oplus 1 \otimes e \oplus \varepsilon \otimes 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \\ 5 \end{pmatrix}.$$

The set of $n \times n$ matrices endowed with these two operations is also a dioid which is denoted by $S^{n \times n}$ (COHEN *et al.*, 1989, Sec. 2.3). The identity matrix of $S^{n \times n}$, denoted by $e^{n \times n}$, has entries equal e on the diagonal and ε elsewhere. The null matrix, denoted by $\varepsilon^{n \times n}$, has all entries equal ε.

The k^{th} power of $A \in S^{n \times n}$ is denoted by $A^{\otimes k}$, or equivalently A^k, and corresponds to $A^k = A^{k-1} \otimes A$. It should be noted that A^0 corresponds to the identity matrix $e^{n \times n}$. Moreover, the star operation is given by:

$$A^* = \bigoplus_{k \in \mathbb{N}} A^k. \tag{2.10}$$

The partial order relation in $S^{n \times n}$ is defined as:

$$A \succeq B \Longleftrightarrow \{a_{ij} \succeq b_{ij}, \; \forall i, j\}. \tag{2.11}$$

Since addition of matrices simply involves the addition of similar entries, $S^{n \times n}$ is complete whenever S is so (COHEN *et al.*, 1989, Sec. 2.3). Moreover, if $S^{n \times n}$ is complete, for any $A \in S^{n \times n}$ and $B \in S^{n \times n}$ it follows that:

$$(A \wedge B)_{ij} = a_{ij} \wedge b_{ij}. \tag{2.12}$$

2.2 Linear Equations in Complete Dioids

This section briefly review some basic concepts on solving linear equations in complete dioids (BACCELLI *et al.*, 1992) (COHEN *et al.*, 1989). The most general system of linear equations in a dioid is given by:

$$a \otimes \mathbf{x} \oplus b = c \otimes \mathbf{x} \oplus d, \tag{2.13}$$

where $a,\ b,\ c,\ d \in S$ and $\mathbf{x} \in S$ is the unknown of the equation. The dioid S is assumed to be complete.

We are especially interested in a subclasse of this general equation given by:

$$a \otimes \mathbf{x} \oplus b = d \tag{2.14}$$

Equation (2.14) admits a solution if and only if $b \preceq d$ and, even in this case, existence and uniqueness are not guaranteed. However, if $b \preceq d$, it is possible to find the greatest subsolution of equation (2.14). A subsolution of equation (2.14) is an $\mathbf{x}$ such that $a \otimes \mathbf{x} \oplus b \preceq d$. Moreover, from (COHEN *et al.*, 1989, Theorem 5) we have that, if $\mathbf{x}$ is the greatest subsolution of (2.14) then $\mathbf{x}$ is also the greatest subsolution of :

$$a \otimes \mathbf{x} = d. \tag{2.15}$$

Definition 2.5 (Residuation (COHEN *et al.*, 1989, Def. 7)) *The (left) residue of d by a, denoted by $a \backslash\!\!\!\circ\, d$, is defined as the greatest subsolution of equation* (2.14)*.*

In (COHEN *et al.*, 1989, Theorem 5) it is demonstrated that the following equalities and inequalities hold true.

$$a \otimes (a \backslash\!\!\!\circ\, b) \leq b \tag{2.16}$$

$$a \backslash\!\!\!\circ\, a \geq e \tag{2.17}$$

$$a \otimes (a \backslash\!\!\!\circ\, a) = a \tag{2.18}$$

$$e \backslash\!\!\!\circ\, a = a \tag{2.19}$$

$$\varepsilon \backslash\!\!\!\circ\, a = \infty \tag{2.20}$$

$$(a \backslash\!\!\!\circ\, b) \otimes c \leq a \backslash\!\!\!\circ\, (bc) \tag{2.21}$$

$$a \backslash\!\!\!\circ\, (b \backslash\!\!\!\circ\, c) = (b \otimes a) \backslash\!\!\!\circ\, c \tag{2.22}$$

$$(a \backslash\!\!\!\circ\, b) \oplus (a \backslash\!\!\!\circ\, c) \leq a \backslash\!\!\!\circ\, (b \oplus c) \tag{2.23}$$

$$(a \backslash\!\!\!\circ\, b) \oplus (c \backslash\!\!\!\circ\, b) \leq (a \wedge c) \backslash\!\!\!\circ\, b \tag{2.24}$$

$$(a \backslash\!\!\!\circ\, b) \wedge (c \backslash\!\!\!\circ\, b) = (a \oplus c) \backslash\!\!\!\circ\, b \tag{2.25}$$

$$(a \backslash\!\!\!\circ\, b) \wedge (a \backslash\!\!\!\circ\, c) = a \backslash\!\!\!\circ\, (b \wedge c) \tag{2.26}$$

The operator $\diamond\!\!\!\backslash$ can be extended to matrices (see (BACCELLI *et al.*, 1992, Lemma 4.83)). Let $A \in S^{n\times p}$ and $B \in S^{n\times m}$, then:

$$(A\diamond\!\!\!\backslash B)_{ij} = \bigwedge_{k=1}^{n} a_{ki}\diamond\!\!\!\backslash b_{kj}. \tag{2.27}$$

Remark 2.6 *Note that computing $A\diamond\!\!\!\backslash B$ corresponds to perform a kind of matrix product $A^T \odot B$, where A^T is the transpose of A and $\odot$ is a new matrix product where the operations $\oplus$ and $\otimes$ are replaced by $\wedge$ and $\diamond\!\!\!\backslash$, respectively (COHEN* et al.*, 1989, Theorem 8).*

Therefore, the system of linear equations given by:

$$A \otimes \mathbf{x} = \mathbf{b}, \tag{2.28}$$

where $A \in S^{n\times p}$ and $\mathbf{b} \in S^{n\times 1}$, admits a greatest subsolution given by $A\diamond\!\!\!\backslash\mathbf{b}$.

2.3 Max-Plus Linear Systems

The Max-Plus Linear (MPL) systems are discrete-event dynamic systems with continuous state space representing the *dates* of occurrence of the events involved in the system modeling. The MPL systems are subject to *synchronization phenomena* and described in terms of "linear" equations in the max-plus semiring (or max-plus algebra) (BACCELLI *et al.*, 1992, Chap. 3). The max- plus semiring, noted by $\overline{\mathbb{R}}_{max}$, is a complete idempotent semiring and is defined as the set $\mathbb{R} \cup \{-\infty\} \cup \{\infty\}$ and the operations:

$$a \oplus b \equiv \max\{a, b\}. \tag{2.29}$$

$$a \otimes b \equiv a + b. \tag{2.30}$$

Moreover, the operations $\wedge$ and $\diamond\!\!\!\backslash$ are defined as follows:

$$a \wedge b \equiv \min\{a, b\}, \tag{2.31}$$

$$a\diamond\!\!\!\backslash b \equiv b - a. \tag{2.32}$$

The identity and the zero element of the Max-Plus semiring are, respectively, $e = 0$ and $\varepsilon = -\infty$, the top element is $\top = \infty$. According to (2.4), in this algebraic structure, a partial order relation is defined by:

$$a \succeq b \Leftrightarrow a = a \oplus b \Leftrightarrow b = a \wedge b. \tag{2.33}$$

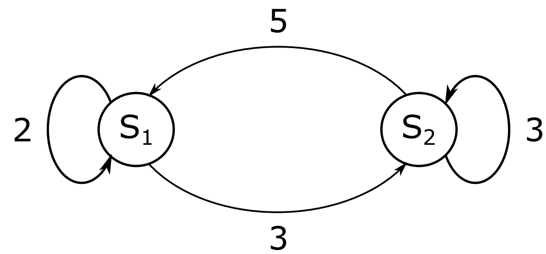

Figure 1 – Railway network model (precedence graph).

Remark 2.7 *Note that $\overline{\mathbb{R}}_{max}$ is linearly ordered with respect to $\oplus$ and the order $\succeq$ in $\overline{\mathbb{R}}_{max}$ coincides with the usual linear order $\geq$ (LITVINOV; SOBOLEVSKIĬ, 2001).*

The basic max-plus operations can be extended to matrices as presented in (2.8), (2.9), (2.12) and (2.27).

The autonomous model of an MPL system is given by:

$$\mathbf{x}(k) = A \otimes \mathbf{x}(k-1), \tag{2.34}$$

where $A \in \overline{\mathbb{R}}^{n \times n}_{max}$ is a matrix that represents the minimal delay between two events. The entries of A are the parameters of the model. The variable $k \in \mathbb{N}$ is an event-number and the state vector $\mathbf{x} \in \overline{\mathbb{R}}^{n}_{max}$ is a **dater**, i.e, $\mathbf{x}(k)$ contains the k-th date of occurrence of each event of the system.

The MPL systems are used to model a wide range of discrete-event systems subject to synchronization phenomena, such as, manufacturing systems, telecommunication networks, railway networks, and parallel computing (BACCELLI *et al.*, 1992, Sec. 1.2).

Example 2.8 (see (CASSANDRAS *et al.*, 1995, Sec. 0.1)) *Consider a public transportation system consisting of two stations S_1 and S_2 and four rail tracks. The structure of the system is given in Figure 1. It is assumed that the train company operates one train on each track initially; the travel times are fixed as indicated on the arcs; trains scheduled to depart must wait for all arriving trains before departing to allow for changeover of passengers; and departures occur as soon as possible. Thus, departures from a station S_i, $i = \{1, 2\}$ will occur at the same time, denoted by $x_i(k)$. The first departure times are assumed to be known and given by $\boldsymbol{x}(0)$. The k-th departure times are given by $\boldsymbol{x}(k)$, where $\boldsymbol{x}(k) = (x_1(k)\ x_2(k))^T$.*

Given these conditions, departures from S_1 must wait for the train arriving from the same station, which takes 2 time units of time, as well as the train arriving from S_2, which takes 5 units of time. Similarly, departures from S_2 must wait for the train arriving from the same station as well as the train arriving from S_1. Therefore, the earliest departure times are *given by:*

$$\begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} = \begin{pmatrix} \max\{2 + x_1(k-1),\ 5 + x_2(k-1)\} \\ \max\{3 + x_1(k-1),\ 3 + x_2(k-1)\} \end{pmatrix}.$$

This system is nonlinear in the conventional algebra, however it can be expressed as the following linear system in the max-plus algebra.

$$\begin{aligned} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} &= \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes \begin{pmatrix} x_1(k-1) \\ x_2(k-1) \end{pmatrix}, \\ &= A \otimes \boldsymbol{x}(k-1). \end{aligned} \tag{2.35}$$

The nonautonomous model of an MPL system is defined by considering an external input $\mathbf{u}$ in (2.34):

$$\mathbf{x}(k) = A \otimes \mathbf{x}(k-1) \oplus B \otimes \mathbf{u}(k), \tag{2.36}$$

where $A \in \overline{\mathbb{R}}_{max}^{n\times n}$, $B \in \overline{\mathbb{R}}_{max}^{n\times m}$.

A nonautonomous MPL system can be transformed into an augmented autonomous MPL model by considering $F = (A\ B) \in \bar{R}_{max}^{n\times(n+m)}$ and $\mathbf{y}(k-1) = \left(\mathbf{x}(k-1)^T\ \mathbf{u}(k)^T\right)^T$ (BACCELLI *et al.*, 1992, Sec. 2.5.4).

$$\mathbf{x}(k) = F \otimes \mathbf{y}(k-1). \tag{2.37}$$

Example 2.9 *Consider a nonautonomous MPL system given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 3 & 2 & 2 \\ e & 1 & 3 \\ 2 & 1 & e \end{pmatrix} \otimes \boldsymbol{x}(k-1) \oplus \begin{pmatrix} e & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \end{pmatrix} \otimes \boldsymbol{u}(k),$$

where $\boldsymbol{x}(k) \in \overline{\mathbb{R}}_{max}^3$ *and* $\boldsymbol{u}(k) \in \overline{\mathbb{R}}_{max}^2$.

The corresponding augmented autonomous MPL model is given by:

$$\boldsymbol{x}(k) = \begin{pmatrix} 3 & 2 & 2 & e & \varepsilon \\ e & 1 & 3 & \varepsilon & e \\ 2 & 1 & e & \varepsilon & \varepsilon \end{pmatrix} v\boldsymbol{y}(k-1),$$

where $\boldsymbol{y}(k-1) = [x_1(k-1)\ x_2(k-1)\ x_3(k-1)\ u_1(k)\ u_2(k)]^T \in \overline{\mathbb{R}}_{max}^5$.

In the following, the classical concepts of eigenvalue and eigenvector are exported to max-plus systems (BACCELLI *et al.*, 1992, Sec. 3.2.4), i.e., given a matrix $A \in \overline{\mathbb{R}}_{max}^{n\times n}$ we consider the problem of existence of eigenvalues λ and eigenvectors ξ such that:

$$A \otimes \xi = \lambda \otimes \xi. \tag{2.38}$$

The solution of this problem depends on the notion of matrix irreducibility, which follows from the definition of precedence graph and strongly connected graph. Moreover, we present the notions of critical graph and cyclicity of a graph.

Definition 2.10 *For a matrix $A \in \overline{\mathbb{R}}_{max}^{n\times n}$, the following notions are defined:*

> **Precedence graph***: The precedence graph of a matrix A is a weighted directed graph with vertices $1, ..., n$ and an arc (j, i) with weight a_{ij} for each $a_{ij} \neq \varepsilon$ (BACCELLI* et al.*, 1992, Def. 2.8).*
>
> **Strongly connected graph***: The precedence graph of A is called strongly connected if for any two different nodes i and j there exists a path from i to j (BACCELLI* et al.*, 1992, Sec. 2.2).*
>
> **Irreducible matrix***: The matrix A is called irreducible if its precedence graph is strongly connected (BACCELLI* et al.*, 1992, Th. 2.14).*
>
> **Length of a path***: A path in a graph is a sequence of nodes $(i_1 \to i_2 \to \cdots \to i_k)$. The length of a path is equal to the sum (in the classical algebra) of the lengths of the arcs of which it is composed, the lengths of the arcs being 1 unless otherwise specified (BACCELLI* et al.*, 1992, Sec. 2.2).*
>
> **Cycle mean***: The mean weight of a path in the precedence graph of A is defined as the sum of the weight of the individual arcs of this path, divided by the length of this path. If such a path is a circuit $(i_1 \to i_2 \to \cdots \to i_1)$ one talks about the mean weight of the circuit, or simply the cycle mean (BACCELLI* et al.*, 1992, Def 2.18). The maximum of these cycle means is called maximum cycle mean. All the operations are in the classical algebra.*
>
> **Critical circuit***: A circuit of the precedence graph of A is critical if its mean weight attains the maximum cycle mean in the precedence graph of A (BACCELLI* et al.*, 1992, Def. 3.94).*
>
> **Critical graph***: The critical graph of A consists of those nodes and arcs which belong to a critical circuit of the precedence graph of A, the weights are set to be equal to e (BACCELLI* et al.*, 1992, Def. 3.94).*

Cyclicity: *The cyclicity of a strongly connected graph is the greatest common divisor g.c.d of the lengths of all its circuits. The cyclicity of a general graph is the least common multiple of the cyclicities of all its strongly connected subgraphs (BACCELLI* et al.*, 1992, Def. 3.94).*

Proposition 2.11 (see (BACCELLI *et al.*, 1992, Th. 3.23)) *If $A \in \overline{\mathbb{R}}_{max}^{n\times n}$ is irreducible there exists one and only one eigenvalue (but possibly several eigenvectors). This eigenvalue corresponds to the maximum cycle mean of the precedence graph of A and is equal to:*

$$\lambda = \bigoplus_{j=1}^{n} \Big(trace(A^j) \Big)^{1/j}. \tag{2.39}$$

Where, for any $B \in \overline{\mathbb{R}}_{max}^{n\times n}$ and $a \in \overline{\mathbb{R}}_{max}$:

$$trace(B) = \bigoplus_{i=1}^{n} b_{ii}, \quad (a^j)^{1/j} = a.$$

The following result can be found in the proof of (BACCELLI *et al.*, 1992, Th. 3.23).

Proposition 2.12 *Let $A \in \overline{\mathbb{R}}_{max}^{n\times n}$ be an irreducible matrix and define $B = \lambda^{-1} \otimes A$ and $B^+ = B \otimes B^*$, where λ is the eigenvalue of A. Then, the matrix B^+ has at least one column with diagonal entry equal to e (the maximum circuit weight in the precedence graph of B is e) and this (these) column(s) is (are) eigenvector(s) of A corresponding to the eigenvalue λ. The set of all eigenvectors corresponding to the eigenvalue λ is the eigenspace noted by $E(A) = \{x \in \overline{\mathbb{R}}_{max}^{n} : A \otimes x = \lambda \otimes x\}$.*

It should be noted that, given a matrix $A \in \overline{\mathbb{R}}_{max}^{n\times n}$ with maximum cycle mean λ, the matrix $B = \lambda^{-1} \otimes A$ (which corresponds to $B = -\lambda + A$ in the classical algebra) has maximum cycle mean equal to e. Therefore, since there are no circuits in the precedence graph of B with positive weight, the existence of B^* is guaranteed (see (BACCELLI *et al.*, 1992, Th. 3.20)).

Proposition 2.13 follows from the cyclicity theorem of the max-plus algebra (BACCELLI *et al.*, 1992, Sec. 3.7), (GAUBERT; PLUS, 1997, Th. 14), (HEIDERGOTT *et al.*, 2006, Th. 3.9).

Proposition 2.13 *Let $A \in \overline{\mathbb{R}}_{max}^{n\times n}$ be an irreducible matrix. There is an integer $K_0(A)$ such that:*

$$k \geq K_0(A) \Rightarrow A^{k+c} = \lambda^c A^k, \tag{2.40}$$

where c is the cyclicity of the critical graph of A and λ is the eigenvalue of A. The smallest $K_0(A)$ verifying this proposition is called the transient time of A.

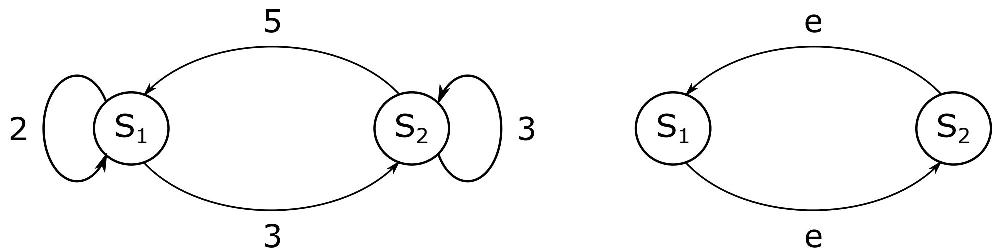

Figure 2 – The precedence graph of A (left) and corresponding critical graph (right).

Proposition 2.13 implies the existence of a periodic behavior of an MPL system.

Corollary 2.14 (see (HEIDERGOTT *et al.*, 2006, Sec. 3.1)) *Given an MPL system characterized by an irreducible matrix $A \in \overline{\mathbb{R}}^{n\times n}_{max}$ and an initial conditions $\boldsymbol{x}(0)$, there exists a finite integer $k_0(\boldsymbol{x}(0))$ such that:*

$$k \geq k_0(\boldsymbol{x}(0)) \Rightarrow \boldsymbol{x}(k+c) = \lambda^c \boldsymbol{x}(k), \tag{2.41}$$

where c is the cyclicity of the critical graph of A and λ is the eigenvalue of A.

Remark 2.15 *Notice that for a given set of initial conditions $\boldsymbol{x}(0)$, it is possible to seek for a specific length of the transient part $k_0(x(0))$, which is, in general, less conservative than the global $K_0(A)$, i.e, $k_0(x(0)) \leq K_0(A)$.*

Example 2.16 *In Example 2.8 we described the railway network model as a MPL system $\boldsymbol{x}(k) = A \otimes \boldsymbol{x}(k-1)$, with*

$$A = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix}.$$

In Figure 2 we recall the precedence graph of A and present the corresponding critical graph. According to Definition 2.10, the precedence graph of A is strongly connected, and therefore the matrix A is irreducible. The precedence graph of A has three circuits: $(S_1 \rightarrow S_1)$ with length 1, $(S_1 \rightarrow S_2 \rightarrow S_1)$ with length 2 and $(S_2 \rightarrow S_2)$ with length 1. Thus, the cyclicity of A is given by $g.c.d(1,2,1) = 1$. The critical graph of A has one circuit $(S_1 \rightarrow S_2 \rightarrow S_1)$ with length equal to 2. Therefore the cyclicity of the critical graph of A is $c = 2$.

The maximum cycle mean of the precedence graph of A, or equivalently, the eigenvalue of A is given by (2.39):

$$\lambda = \bigoplus_{j=1}^{2} \left(trace(A^j)\right)^{1/j} = trace(A)^1 \oplus (trace(A^2))^{1/2},$$

since

$$A^2 = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} \otimes \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} = \begin{pmatrix} 8 & 8 \\ 6 & 8 \end{pmatrix},$$

we have that:

$$\lambda = (2 \oplus 3)^1 \oplus (8 \oplus 8)^{1/2} = 3 \oplus (8)^{1/2} = 3 \oplus 4 = 4.$$

According to Corollary 2.14, it follows that there exists a $K_0(A)$ such that:

$$k \geq K_0(A) \Rightarrow \boldsymbol{x}(k+2) = 4^2 \otimes \boldsymbol{x}(k) = 8 \otimes \boldsymbol{x}(k).$$

Indeed, for $\boldsymbol{x}(0) = (e\ e)^T$, the following sequence $\boldsymbol{x}(k)$, $k = 1, 2, \cdots$ is observed:

$$\begin{pmatrix} e \\ e \end{pmatrix}, \begin{pmatrix} 5 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 8 \end{pmatrix}, \begin{pmatrix} 13 \\ 11 \end{pmatrix}, \begin{pmatrix} 16 \\ 16 \end{pmatrix}, \begin{pmatrix} 21 \\ 19 \end{pmatrix}, \begin{pmatrix} 24 \\ 24 \end{pmatrix}, \begin{pmatrix} 29 \\ 27 \end{pmatrix}, \cdots$$

Therefore, one can conclude that for all $k \geq 0$, $\boldsymbol{x}(k+2) = 8x(k)$, $x(0) = [e\ e]^T$ and $x(1) = [5\ 3]^T$.

To calculate the eigenvector(s) corresponding to the eigenvalue $\lambda = 4$ we define the matrix:

$$B = \lambda^{-1} \otimes A = 4^{-1} \otimes \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} = -4 \otimes \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ -1 & -1 \end{pmatrix}.$$

The matrices B^* and B^+ are:

$$\begin{aligned} B^* &= e \oplus B = \begin{pmatrix} e & \varepsilon \\ \varepsilon & e \end{pmatrix} \oplus \begin{pmatrix} -2 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} e & 1 \\ -1 & e \end{pmatrix}. \\ B^+ &= B \otimes B^* = \begin{pmatrix} -2 & 1 \\ -1 & -1 \end{pmatrix} \otimes \begin{pmatrix} e & 1 \\ -1 & e \end{pmatrix} = \begin{pmatrix} e & 1 \\ -1 & e \end{pmatrix}. \end{aligned}$$

From Proposition 2.12 it follows that $\xi = [e\ -1]^T$ and $\xi = [1\ e]^T$ are eigenvectors of A corresponding to the eigenvalue $\lambda = 4$. According to (2.38), if we set the initial conditions of the system to be equal to an eigenvector of A, the periodic behavior of the system will be given by $\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) = \lambda\boldsymbol{x}(k) = 4\boldsymbol{x}(k)$ for all $k \geq 0$. Indeed, for $\boldsymbol{x}(0) = [1\ e]^T$ the following sequence is observed:

$$\begin{pmatrix} 1 \\ e \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \end{pmatrix}, \begin{pmatrix} 9 \\ 8 \end{pmatrix}, \begin{pmatrix} 13 \\ 12 \end{pmatrix}, \begin{pmatrix} 17 \\ 16 \end{pmatrix}, \begin{pmatrix} 21 \\ 20 \end{pmatrix}, \begin{pmatrix} 25 \\ 24 \end{pmatrix}, \begin{pmatrix} 29 \\ 28 \end{pmatrix}, \cdots$$

2.4 Difference Bounds Matrix

The Difference Bounds Matrices (DBM) are an effective data structure to represent regions defined by a finitely many number of linear inequalities (DILL, 1990).

The DBM are square matrices with entries in the complete idempotent semiring noted by $\mathcal{B}$ (*bounds algebra*) and defined as the set of ordered pairs $(\mathbb{R},\ \bowtie) \cup (\infty,\ <) \cup (-\infty,\ <)$ (where $\bowtie \in \{<,\ \leq\}$ and $<$ is assumed to be strictly less than $\leq$) and the operations of sum and product defined, respectively, as the intersection and sum of the usual algebra:

$$(a, \bowtie_a) \oplus_{\mathcal{B}} (b, \bowtie_b) = \begin{cases} (a, \bowtie_a) & \text{if } a < b \text{ or } (a = b \text{ and } \bowtie_a \leq \bowtie_b), \\ (b, \bowtie_b) & \text{otherwise.} \end{cases} \tag{2.42}$$

$$(a, \bowtie_a) \otimes_{\mathcal{B}} (b, \bowtie_b) = (a + b, min(\bowtie_a, \bowtie_b)). \tag{2.43}$$

The identity and the zero element in $\mathcal{B}$ are, respectively, $e_{\mathcal{B}} = (0, \leq)$ and $\varepsilon_{\mathcal{B}} = (\infty, <)$, the top element is $\top_{\mathcal{B}} = (-\infty, <)$.

According to (2.4), in this algebraic structure, a partial order relation is defined by:

$$(a, \bowtie_a) \succeq_{\mathcal{B}} (b, \bowtie_b) \Leftrightarrow (a, \bowtie_a) = (a, \bowtie_a) \oplus_{\mathcal{B}} (b, \bowtie_b) \Leftrightarrow (b, \bowtie_b) = (a, \bowtie_a) \wedge (b, \bowtie_b). \tag{2.44}$$

Remark 2.17 *The order $\succeq_{\mathcal{B}}$ in $\mathcal{B}$ coincides with the usual lexicographic order $\leq$ (DILL, 1990, Sec. 3.1), i.e.,*

$$(a, \bowtie_a) \succeq_{\mathcal{B}} (b, \bowtie_b) \Leftrightarrow (a, \bowtie_a) \leq (b, \bowtie_b).$$

Equivalently, the order $\preceq_{\mathcal{B}}$ coincides with $\geq$.

Example 2.18 *Consider the sets $A = \{x \in \mathbb{R} : x \leq 3\}$, $B = \{x \in \mathbb{R} : x < 4\}$ and $C = \{x \in \mathbb{R} : x < 3\}$. These sets can be represented, respectively, by the following elements in the bounds algebra: $a = (3, \leq)$, $b = (4, <)$ and $c = (3, <)$. Thus, we have that $c \succeq_{\mathcal{B}} a \succeq_{\mathcal{B}} b$, and*

$$\begin{aligned} A \cap B = \{x \in \mathbb{R} : x \leq 3\} &\equiv a \oplus_{\mathcal{B}} b = (3, \leq), \\ A \cap C = \{x \in \mathbb{R} : x < 3\} &\equiv a \oplus_{\mathcal{B}} c = (3, <), \\ B \cap C = \{x \in \mathbb{R} : x < 3\} &\equiv b \oplus_{\mathcal{B}} c = (3, <), \\ A + B = \{x \in \mathbb{R} : x < 7\} &\equiv a \otimes_{\mathcal{B}} b = (7, <), \\ A + C = \{x \in \mathbb{R} : x < 6\} &\equiv a \otimes_{\mathcal{B}} c = (6, <), \\ B + C = \{x \in \mathbb{R} : x < 7\} &\equiv b \otimes_{\mathcal{B}} c = (7, <). \end{aligned}$$

The star operation is given by:

$$(a, \bowtie_a)^* = e_{\mathcal{B}} \oplus_{\mathcal{B}} (a, \bowtie_a) \oplus_{\mathcal{B}} (a, \bowtie_a)^2 ... = \begin{cases} e_{\mathcal{B}} & \text{if } (a, \bowtie_a) \preceq_{\mathcal{B}} e_{\mathcal{B}}, \\ \top_{\mathcal{B}} & \text{otherwise.} \end{cases} \tag{2.45}$$

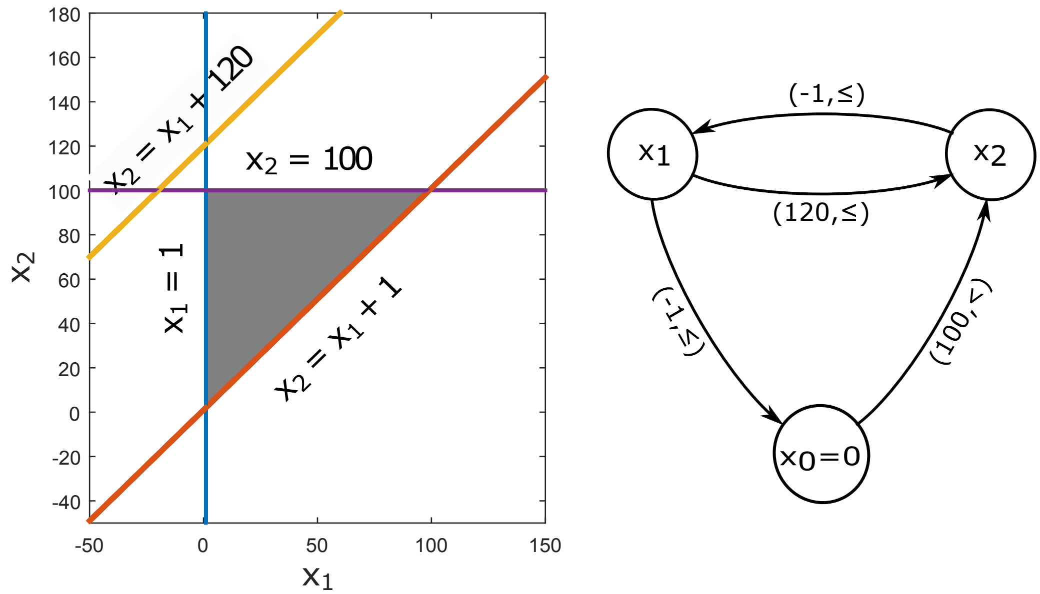

Figure 3 – Region (left) and directed graph representation (right) of D .

A DBM is a square matrix $D \in \mathcal{B}^{n+1\times n+1}$, with diagonal entries $e_{\mathcal{B}}$, representing a system of linear inequalities that constrain single variables in a set $\{x_1,\ x_2, ..., x_n\}$ and their differences within the limits identified by $d_{i+1\ j+1} = (\alpha_{ij}, \bowtie_{ij})$ (DILL, 1990, Sec. 4.1), (RIDI *et al.*, 2012):

$$\begin{cases} x_i - x_j \bowtie_{ij} \alpha_{ij} \\ x_0 = 0 \end{cases} \quad i \neq j \text{ and } i, j \in \{0, ..., n\}\,. \tag{2.46}$$

The artificial value x_0 is assumed to be always equal 0 and is used to represent bounds over a single variable, e.g., $x_i \leq \alpha_{i,0} \Leftrightarrow x_i - x_0 \leq \alpha_{i,0}$ or $x_i \geq -\alpha_{0,i} \Leftrightarrow x_0 - x_i \leq \alpha_{0,i}$. The solution set of (2.46) is the **region** of D, or $\mathcal{R}(D)$.

The identity DBM in $\mathcal{B}^{n\times n}$, denoted by $e_{\mathcal{B}}^{n\times n}$, has entries equal $(0, \leq)$ on the diagonal and $(\infty, <)$ elsewhere. The null matrix, denoted by $\varepsilon_{\mathcal{B}}^{n\times n}$, has all entries equal $(\infty, <)$.

Remark 2.19 *We can also look at a DBM as a directed graph in which inequality bounds become arc weights.*

Example 2.20 *Consider the following DBM:*

$$D = \begin{pmatrix} e_{\mathcal{B}} & (-1, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-1, \leq) \\ (100, <) & (120, \leq) & e_{\mathcal{B}} \end{pmatrix}$$

The region of D is given by $\mathcal{R}(D) = \{\boldsymbol{x} \in \mathbb{R} : x_1 \geq 1,\ 1 \leq x_2 - x_1 \leq 120,\ x_2 < 100\}$ as presented in Figure 3.

Given two DBM in $\mathcal{B}^{n\times n}$, $D^{(X)}$ and $D^{(Y)}$, according to (2.11) the partial order relation can be defined as:

$$D^{(X)} \succeq_{\mathcal{B}} D^{(Y)} \iff D^{(X)} = D^{(X)} \oplus_{\mathcal{B}} D^{(Y)} \iff \{d_{ij}^{(A)} \succeq d_{ij}^{(B)},\ \forall i,j\}. \tag{2.47}$$

Remark 2.21 *The sum (in $\mathcal{B}$) of DBM is equivalent to the intersection of its regions, e.g, let $D^{(X)}$ and $D^{(Y)}$ be two DBM in $\mathcal{B}^{n\times n}$. Then, $D^{(X)} \oplus_{\mathcal{B}} D^{(Y)} \equiv \mathcal{R}(D^{(X)}) \cap \mathcal{R}(D^{(Y)})$. From now on, the sum of DBM will be referred as the intersection of DBM.*

Remark 2.22 *In general, the union of DBM is not a DBM. However, if $D^{(X)} = D^{(X)} \oplus_{\mathcal{B}} D^{(Y)}$ then $D^{(X)} \cup D^{(Y)} = D^{(Y)}$. Note that $D^{(X)} \oplus_{\mathcal{B}} D^{(Y)}$ corresponds to the intersection of $D^{(X)}$ and $D^{(Y)}$.*

2.4.1 Canonical Form Representation and Checking for Emptiness

In general, a region can be represented by several DBM. However, each DBM admits an equivalent and unique representation in canonical form, given by (DILL, 1990, Th. 2):

$$cf(D) = D^*. \tag{2.48}$$

By definition $D^*[i,j]$ is the cost of the shortest path[1] in the precedence graph of D from node with index i to j (DILL, 1990, Sec. 4.1). Therefore, the Floyd-Warshall algorithm (FLOYD, 1962) (see also algorithm 2.1) can be used to obtain the canonical-form representation of a DBM with a complexity that is cubic w.r.t. its dimension. Note that, if there is a cycle of cost less than $(0, \leq)$ in the precedence graph of a given DBM D, a path of arbitrarily small cost can be obtained by repeating the negative cost cycle. In the limit we would obtain $D^*[i,j] = (-\infty, <) \Rightarrow x_i - x_j < -\infty$, for some $(i,j) \in \{0, \cdots, n\} \times \{0, \cdots, n\}$, and therefore the system represented by D is inconsistent, or equivalently $\mathcal{R}(D) = \emptyset$. Thus, a simple way to decide if D has empty region is to check if a negative-cost cycle appears during the computation of the shortest-path matrix using the Floyd-Warshall algorithm (DILL, 1990, Sec. 4.1).

Algorithm 2.1 presents the Floyd-Warshall algorithm with a checking for emptiness step. The algorithm works as follows: at the first iteration, it is computed the shortest path among all pairs of nodes with the restriction that only the node with index 0 can be visited

[1] The longest path in $\mathcal{B}$ (see remark 2.17)

as intermediary nodes; at the second iteration, it is computed the shortest path among all pairs of nodes with the restriction that only nodes with index in $\{0, 1\}$ can be visited as intermediary nodes. Finally, at the n-th iteration, it is computed the shortest path among all pairs of nodes using any node in the precedence graph of D as intermediary node. Note that step 6 checks for negative-cost cycles. If a negative-cost cycle is detected, $D^*[1, 1]$ is actualized with the value $\top_{\mathcal{B}} = (-\infty, <)$ to signalizes that D has empty region and the algorithm is stopped.

Algorithm 2.1: Floyd-Warshall algorithm (operations in $\mathcal{B}$).

input : $D \in \mathcal{B}^{n+1 \times n+1}$
output: D^*

1 $D^* \leftarrow D$;
2 **for** $k = 1 \rightarrow n + 1$ **do**
3   **for** $i = 1 \rightarrow n + 1$ **do**
4   **for** $j = 1 \rightarrow n + 1$ **do**
5             $D^*[i, j] \leftarrow D^*[i, j] \oplus_{\mathcal{B}} (D^*[i, k] \otimes_{\mathcal{B}} D^*[k, j])$;
6   **if** $i == j$ *and* $D^*[i, j] \succ_{\mathcal{B}} e_{\mathcal{B}}$ **then**
7                 $D^*[1, 1] \leftarrow \top_{\mathcal{B}}$, $k \leftarrow n$, $i \leftarrow n$, $j \leftarrow n$;
8   **end**
9   **end**
10   **end**
11 **end**

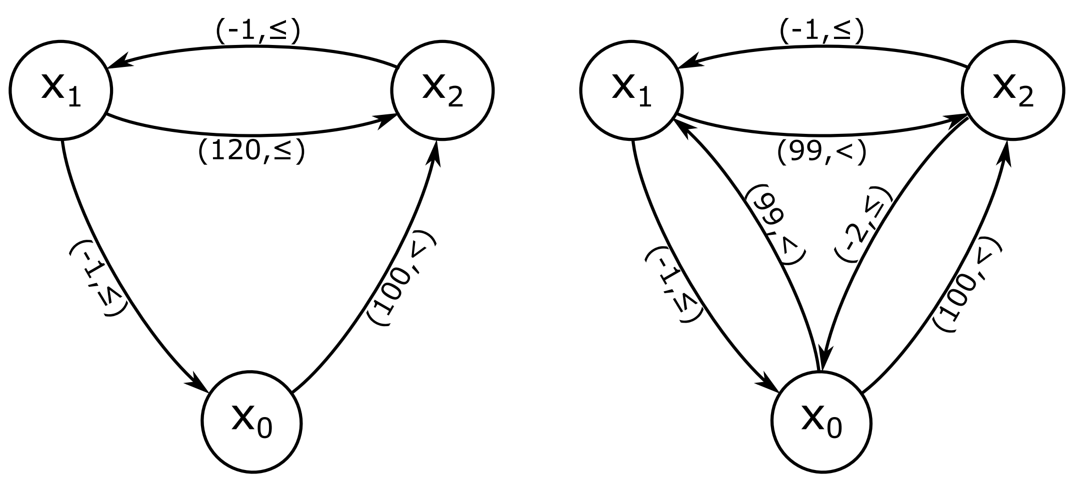

Figure 4 – Directed graph representation of D (left) and its canonical form (right).

Example 2.23 *The canonical form representation of the DBM of example 2.20 is given by:*

$$D^* = \begin{pmatrix} e_{\mathcal{B}} & (-1, \leq) & (-2, \leq) \\ (99, <) & e_{\mathcal{B}} & (-1, \leq) \\ (100, <) & (99, <) & e_{\mathcal{B}} \end{pmatrix}.$$

In figure 4, are the precedence graphs of D and D^*.

Definition 2.24 (stripe) *A **stripe** is defined as a DBM $D \in \mathcal{B}^{(n)\times(n)}$, whose canonical form representation D^* is such that $D^*[1,j] = D^*[j,1] = \varepsilon_{\mathcal{B}}$ for all $j \in \{1, ..., n\}$.*

Remark 2.25 *Note that, according to Definition 2.24, a stripe is a DBM that does not constrain single variables, and therefore does not require the artificial variable x_0. In (ADZKIYA* et al.*, 2015, Sec. 2.3) a stripe is defined as a DBM that does mot contain the variable x_0.*

2.4.2 Orthogonal Projection and Cartesian Product of DBM

This section presents two important operations with DBM: the Orthogonal Projection onto a subset of its variables and the Cartesian (or cross) product of DBM.

Given a DBM $D \in \mathcal{B}^{n\times n}$, which constrain the variables $\{x_1, ..., x_n\}$ and their differences, the orthogonal projection of D onto a subset $\{x_{i_1}, ..., x_{i_p}\}$, written $D\lceil_{\{x_{i_1},...,x_{i_p}\}}$, is such that $\mathcal{R}(D\lceil_{\{x_{i_1},...,x_{i_p}\}}) = \{(x_{i_1}, ..., x_{i_p})^T \in \mathbb{R}^p : (x_1, ..., x_n)^T \in \mathcal{R}(D)\}$. If the DBM is in the canonical form, its orthogonal projection onto a subset of its variables can be find by deleting the rows and columns corresponding to the complementary variables, i.e, the variables x_j such that $j \notin \{i_1, ..., i_p\}$ (DILL, 1990, Sec. 4.1).

Given two DBM $D^{(X)} \in \mathcal{B}^{(p+1)\times(p+1)}$ and $D^{(Y)} \in \mathcal{B}^{(n+1)\times(n+1)}$, the Cartesian product of its regions is given by $\mathcal{R}(D^{(X)}) \times \mathcal{R}(D^{(Y)}) = \{\left(\mathbf{x}^T, \mathbf{y}^T\right)^T \in \mathbb{R}^{p+n} : \mathbf{x} \in \mathcal{R}(D^{(X)}),\ \mathbf{y} \in \mathcal{R}(D^{(Y)})\}$. From the DBM point of view, the Cartesian product $D^{(X)} \times D^{(Y)}$ can be represented by an augmented DBM $D^{(X\times Y)} \in \mathcal{B}^{(p+n+1)\times(p+n+1)}$ such that $\mathcal{R}(D^{(X\times Y)}) = \mathcal{R}(D^{(X)}) \times \mathcal{R}(D^{(Y)})$. Algorithm 2.2 constructs $D^{(X\times Y)}$ with complexity $\mathcal{O}(n^2)$.

Example 2.26 *Consider the following DBM:*

$$D^{(X)} = \begin{array}{cc} \begin{array}{cc} x_0 & x_1 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} \\ (80, \leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1 \end{array} \end{array} \qquad D^{(Y)} = \begin{array}{cc} \begin{array}{ccc} x_0 & y_1 & y_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & (-1, \leq) & (-2, \leq) \\ (99, <) & e_{\mathcal{B}} & (-1, \leq) \\ (100, <) & (99, <) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ y_1 \\ y_2 \end{array} \end{array}$$

The Cartesian product of the DBM is given by:

Algorithm 2.2: Cartesian product of DBM.

input : $D^{(X)} \in \mathcal{B}^{(p+1)\times(p+1)}$ and $D^{(Y)} \in \mathcal{B}^{(n+1)\times(n+1)}$
output: $D^{(X\times Y)} = (D^{(X)} \times D^{(Y)}) \in \mathcal{B}^{(p+n+1)\times(p+n+1)}$

$D^{(X\times Y)} \leftarrow e_{\mathcal{B}}^{(p+n+1)\times(p+n+1)}$;
for $i = 1 \rightarrow p+1$ **do**
  for $j = 1 \rightarrow p+1$ **do**
    $D^{(X\times Y)}[i,j] \leftarrow D^{(X)}[i,j]$;
  end
end
for $i = 2 \rightarrow n+1$ **do**
  $D^{(X\times Y)}[1, p+i] \leftarrow D^{(Y)}[1,i]$;
  $D^{(X\times Y)}[p+i, 1] \leftarrow D^{(Y)}[i,1]$;
  for $j = 2 \rightarrow n+1$ **do**
    $D^{(X\times Y)}[p+i, p+j] \leftarrow D^{(Y)}[i,j]$;
  end
end

$$D^{(X\times Y)} = D^{(X)} \times D^{(Y)} = \begin{array}{c c} \begin{array}{cccc} x_0 & x_1 & y_1 & y_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) & (-2,\leq) \\ (80,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (99,<) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) \\ (100,<) & \varepsilon_{\mathcal{B}} & (99,<) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1 \\ y_1 \\ y_2 \end{array} \end{array}$$

The canonical form of $D^{(X\times Y)}$ is given by:

$$cf(D^{(X\times Y)}) = \begin{array}{c c} \begin{array}{cccc} x_0 & x_1 & y_1 & y_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) & (-2,\leq) \\ (80,\leq) & e_{\mathcal{B}} & (79,\leq) & (78,\leq) \\ (99,<) & (99,<) & e_{\mathcal{B}} & (-1,\leq) \\ (100,<) & (100,<) & (99,<) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1 \\ y_1 \\ y_2 \end{array} \end{array}$$

The orthogonal projection of $cf(D^{(X\times Y)})$ over the variables $\boldsymbol{x}$ is obtained by deleting the rows and columns corresponding to the variables $\boldsymbol{y}$. Thus

$$cf(D^{(X\times Y)})\lceil_{\boldsymbol{x}} = \begin{array}{c c} \begin{array}{cc} x_0 & x_1 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} \\ (80,\leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1 \end{array} \end{array}$$

Equivalently, the orthogonal projection of $cf(D^{(X\times Y)})$ over the variables $\boldsymbol{y}$ is given by.

$$cf(D^{(X\times Y)})\lceil_{\boldsymbol{y}} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} x_0 & y_1 & y_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-1,\leq) & (-2,\leq) \\ (99,<) & e_{\mathcal{B}} & (-1,\leq) \\ (100,<) & (99,<) & e_{\mathcal{B}} \end{pmatrix} \end{array}\begin{array}{l} \\ x_0 \\ y_1 \\ y_2 \end{array}$$

2.5 Piece-Wise Affine Systems

This section discusses Piece-Wise Affine (PWA) systems generated by a generic (autonomous or nonautonomous) MPL system (ADZKIYA *et al.*, 2015, Sec. 2.2). The PWA systems (SONTAG, 1981) are described by a collection of state space equations associated with a given region of activity, which is given by a finite number of linear inequalities. They can model a large number of physical processes and can approximate nonlinear dynamics with arbitrary accuracy. PWA systems have been studied in (SONTAG, 1981; CHUA; DENG, 1988; VANDENBERGHE *et al.*, 1989; KEVENAAR; LEENAERTS, 1992; JOHANSSON; RANTZER, 1997; BEMPORAD *et al.*, 2000; HEEMELS *et al.*, 2001; JULIAN, 2003; WEN; MA, 2011).

Consider a generic MPL system given by:

$$\mathbf{z}(k) = A \otimes \mathbf{x}(k-1), \tag{2.49}$$

where $A \in \overline{\mathbb{R}}^{n\times p}_{max}$ and $\mathbf{z}$ and $\mathbf{x}$ are vectors of appropriate dimensions.

Remark 2.27 *Equation* (2.49) *is generic in the sense that it can represent either an autonomous MPL system ($p = n$, see* (2.34)*) or an nonautonomous MPL system ($p = n + m$, see* (2.37)*).*

This system can be expressed as a PWA system in the event domain[2] (HEEMELS *et al.*, 2001):

$$\mathbf{z}(k) = A_{\mathbf{g}}\mathbf{x}(k-1) + \mathbf{f}_{\mathbf{g}} \text{ for } \mathbf{x}(k-1) \in R_{\mathbf{g}}, \tag{2.50}$$

where the collection of all $R_{\mathbf{g}}$, $\mathbf{g} = (g_1, \cdots, g_n) \in \{1, ..., p\}^n$, forms a partition of the state space, $\mathbf{f}_{\mathbf{g}}$ is a vector of constants and $A_{\mathbf{g}}$ is a matrix of suitable dimensions.

Each $\mathbf{g}$ is associated with a dynamics and a region $R_{\mathbf{g}}$ such that, for all $\mathbf{x} \in R_{\mathbf{g}}$, the element g_i corresponds to the index of the maximum term of the i-th system equation of (2.49), which can be expressed as:

$$z_i(k) = \bigoplus_{j=1}^{p}\{a_{ij} \otimes x_j(k-1)\}. \tag{2.51}$$

[2] Operations in the classical algebra

Thus,

$$a_{ig_i} \otimes x_{g_i}(k-1) = \bigoplus_{j=1}^{p} \{a_{ij} \otimes x_j(k-1)\}. \tag{2.52}$$

From (2.33), equation (2.52) can be expressed as:

$$a_{ij} + x_j(k-1) \leq a_{ig_i} + x_{g_i}(k-1) \quad \forall j \in \{1, \cdots, p\}. \tag{2.53}$$

Therefore, the region $R_{\mathbf{g}}$ which represents the set of all $\mathbf{x} \in \overline{\mathbb{R}}^p_{max}$ that satisfies (2.53), is given by:

$$R_{\mathbf{g}} = \bigcap_{i=1}^{n} \bigcap_{\substack{j=1 \\ j \neq g_i}}^{p} \left\{ \mathbf{x} \in \overline{\mathbb{R}}^p_{max} : x_j - x_{g_i} \leq a_{ig_i} - a_{ij} \right\}. \tag{2.54}$$

From, (2.52), the affine dynamics that is active in $R_{\mathbf{g}}$ is given by:

$$z_i(k) = x_{g_i}(k-1) + a_{ig_i}, \; i \in \{1, ..., n\}. \tag{2.55}$$

Therefore, the generic MPL system (2.49) can be expressed as the PWA system given in (2.50) where, for each $\mathbf{g}$, the region $R_{\mathbf{g}}$ is given by (2.54), the matrix $A_{\mathbf{g}}$ is such that, for all $(i, j) \in \{1, ..., n\} \times \{1, ..., p\}$:

$$A_{\mathbf{g}}(i,j) = \begin{cases} 1 & \text{if } j = g_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.56}$$

and the vector of constants $\mathbf{f}_{\mathbf{g}}$ is given by:

$$\mathbf{f}_{\mathbf{g}} = \begin{pmatrix} a_{1g_1} \\ a_{2g_2} \\ \cdots \\ a_{ng_n} \end{pmatrix}. \tag{2.57}$$

Example 2.28 *Consider the autonomous MPL system given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 8 & 5 \\ 4 & 3 \end{pmatrix} \otimes \boldsymbol{x}(k-1).$$

According to equation (2.54)*, the regions corresponding to each* $\boldsymbol{g} \in \{1, 2\}^2 = \{(1, 1),$ $(1, 2)$, $(2, 1)$, $(2, 2)\}$ *are given by:*

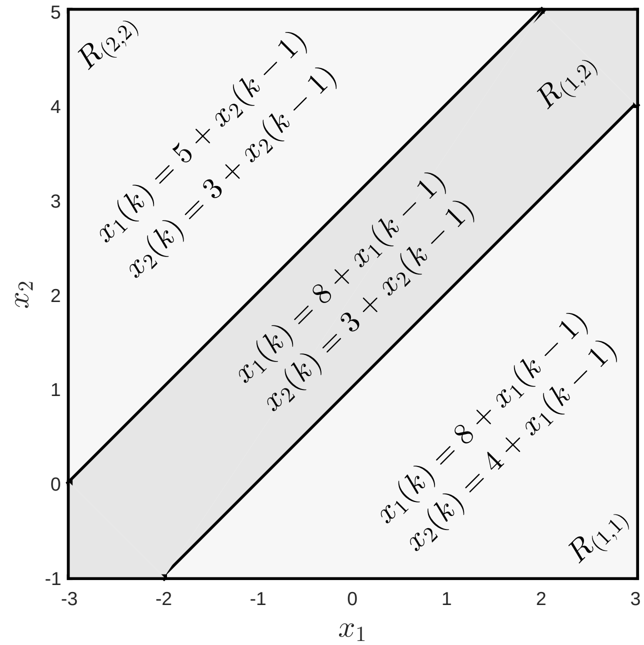

Figure 5 – A PWA system generated by an MPL system.

$$
\begin{aligned}
R_{(1,1)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 3\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 1\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 1\right\}, \\
R_{(1,2)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 3\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -1\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 1 \leq x_2 - x_1 \leq 3\right\}, \\
R_{(2,1)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -3\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq -1\right\} \\
&= \emptyset, \\
R_{(2,2)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -3\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -1\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \geq 3\right\}.
\end{aligned}
$$

Thus, according to equations (2.56) *and* (2.57), *the corresponding PWA system, defined by* (2.50), *is given by:*

$$
\boldsymbol{x}(k) = \begin{cases}
\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \boldsymbol{x}(k-1) + \begin{pmatrix} 8 \\ 4 \end{pmatrix} & \textit{if} \quad \boldsymbol{x}(k-1) \in R_{(1,1)}, \\
\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{x}(k-1) + \begin{pmatrix} 8 \\ 3 \end{pmatrix} & \textit{if} \quad \boldsymbol{x}(k-1) \in R_{(1,2)}, \\
\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \boldsymbol{x}(k-1) + \begin{pmatrix} 5 \\ 3 \end{pmatrix} & \textit{if} \quad \boldsymbol{x}(k-1) \in R_{(2,2)},
\end{cases}
$$

Figure 5 depicts the PWA system generated by A.

Example 2.29 *Consider the nonautonomous MPL system given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 2 & 4 \\ 3 & e \end{pmatrix} \otimes \boldsymbol{x}(k-1) \oplus \begin{pmatrix} e \\ \varepsilon \end{pmatrix} \otimes \boldsymbol{u}(k).$$

According to (2.37), *the corresponding augmented autonomous MPL system is given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 2 & 4 & e \\ 3 & e & \varepsilon \end{pmatrix} \otimes \boldsymbol{y}(k-1), \text{ where } \boldsymbol{y}(k-1) = \begin{pmatrix} x_1(k-1) \\ x_2(k-1) \\ u_1(k) \end{pmatrix}.$$

According to equation (2.54), *the regions corresponding to each* $\boldsymbol{g} \in \{1,2,3\}^2 = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\}$ *are given by:*

$$\begin{aligned}
R_{(1,1)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \le 2\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le 3\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \le \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le -2,\ y_3 - y_1 \le 2\right\},
\end{aligned}$$

$$\begin{aligned}
R_{(1,2)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \le 2\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \le -3\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \le \infty\right\} \\
&= \emptyset,
\end{aligned}$$

$$\begin{aligned}
R_{(1,3)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \le 2\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_3 \le -\infty\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_3 \le -\infty\right\} \\
&= \emptyset,
\end{aligned}$$

$$\begin{aligned}
R_{(2,1)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \le 2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \le 4\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \le 3\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \le \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : -2 \le y_2 - y_1 \le 3,\ y_3 - y_2 \le 4\right\},
\end{aligned}$$

$$\begin{aligned}
R_{(2,2)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \le 2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \le 4\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \le -3\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \le \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \ge 3,\ y_3 - y_2 \le 4\right\},
\end{aligned}$$

$$\begin{aligned}
R_{(2,3)} &= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq 2\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq 4\right\} \\
&\qquad \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -\infty\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -\infty\right\} \\
&= \emptyset,
\end{aligned}$$

$$\begin{aligned}
R_{(3,1)} &= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \\
&\qquad \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq 3\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq \infty\right\} \\
&= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2,\ y_2 - y_3 \leq -4,\ y_2 - y_1 \leq 3\right\},
\end{aligned}$$

$$\begin{aligned}
R_{(3,2)} &= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \\
&\qquad \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -3\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq \infty\right\} \\
&= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2,\ y_2 - y_3 \leq -4,\ y_2 - y_1 \geq 3\right\},
\end{aligned}$$

$$\begin{aligned}
R_{(3,3)} &= \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \\
&\qquad \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -\infty\right\} \cap \left\{\boldsymbol{y}\in\overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -\infty\right\} \\
&= \emptyset.
\end{aligned}$$

Thus, according to equations (2.56) *and* (2.57)*, the corresponding PWA system, defined by* (2.50)*, is given by:*

$$\boldsymbol{x}(k) = \begin{cases}
\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \boldsymbol{y}(k-1) + \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \textit{if} \quad \boldsymbol{y}(k-1) \in R_{(1,1)}, \\
\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \boldsymbol{y}(k-1) + \begin{pmatrix} 4 \\ 3 \end{pmatrix} & \textit{if} \quad \boldsymbol{y}(k-1) \in R_{(2,1)}, \\
\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \boldsymbol{y}(k-1) + \begin{pmatrix} 4 \\ 0 \end{pmatrix} & \textit{if} \quad \boldsymbol{y}(k-1) \in R_{(2,2)}, \\
\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \boldsymbol{y}(k-1) + \begin{pmatrix} 0 \\ 3 \end{pmatrix} & \textit{if} \quad \boldsymbol{y}(k-1) \in R_{(3,1)}, \\
\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \boldsymbol{y}(k-1) + \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \textit{if} \quad \boldsymbol{y}(k-1) \in R_{(3,2)}.
\end{cases}$$

Given a matrix $A \in \overline{\mathbb{R}}^{n\times p}_{max}$, Algorithm 2.3 (taken from (ADZKIYA *et al.*, 2015, Sec. 2.2)) describes a general procedure to generate the corresponding PWA system.

The algorithm works as follows. In step 1, the output variables are initialized. Then, for each $\mathbf{g} \in \{1, ..., p\}^n$ (step 2), the region $R_{\mathbf{g}}$ (step 6), the matrix $A_{\mathbf{g}}$ and the vector $f_{\mathbf{g}}$

Algorithm 2.3: Expressing an MPL system as a PWA system using a bactracking technique. The assignment $zeros(.,.)$ generates a matrix of specified dimensions, with entries equal to 0.

input : $A \in \overline{\mathbb{R}}_{max}^{n \times p}$
output: R, A, f

1 $\mathtt{R} \leftarrow \emptyset$, $\mathtt{A} \leftarrow \emptyset$, $\mathtt{f} \leftarrow \emptyset$;
2 **for all** $\boldsymbol{g} = (g_i, ..., g_n) \in \{1, ..., p\}^n$ **do**
3 $\quad R_{\mathbf{g}} \leftarrow \mathbb{R}^p$, $A_{\mathbf{g}} \leftarrow zeros(n, p)$, $f_{\mathbf{g}} \leftarrow zeros(n, 1)$;
4 $\quad$ **for all** $i \in \{1, ..., n\}$ **do**
5 $\quad\quad$ **for all** $j \in \{1, ..., p\}$ **do**
6 $\quad\quad\quad R_{\mathbf{g}} \leftarrow R_{\mathbf{g}} \cap \{\mathbf{x} \in \mathbb{R}^p : x_j - x_{g_i} \leq a_{ig_i} - a_{ij}\}$; `// define regions` (2.54)
7 $\quad\quad$ **end for**
8 $\quad\quad A_{\mathbf{g}}(i, g_i) \leftarrow 1$, $f_{\mathbf{g}}(i) \leftarrow a_{ig_i}$; `// see equations` (2.56) `and` (2.57)
9 $\quad$ **end for**
10 $\quad$ **if** $R_{\boldsymbol{g}} \neq \emptyset$ **then**
11 $\quad\quad \mathtt{R} \leftarrow \mathtt{R} \cup \{R_{\mathbf{g}}\}$, $\mathtt{A} \leftarrow \mathtt{A} \cup \{A_{\mathbf{g}}\}$, $\mathtt{f} \leftarrow \mathtt{f} \cup \{f_{\mathbf{g}}\}$;
12 $\quad$ **end if**
13 **end for**

(step 8) are constructed according to equations (2.54), (2.56) and (2.57), respectively. If $R_{\mathbf{g}}$ is not empty (step 10), the procedure saves the region and corresponding affine dynamics to the output variables (step 11). The worst-case complexity of the algorithm is $\mathcal{O}\left(p^n(np + p^3)\right)$ (see (ADZKIYA *et al.*, 2015, Sec. 2.3)).

Remark 2.30 *The bottleneck of Algorithm 2.3 resides in the worst-case cardinality of the collection of regions $R_{\boldsymbol{g}}$, given by p^n. Practically, each row i of an $n \times p$ matrix has $p'_i \leq p$ non-ε elements, thus the worst-case cardinality reduces to $\prod_{i=1}^{n} p'_i \leq p^n$. Besides, as many regions can be empty, the complexity of the algorithm is often drastically smaller than the worst-case bound. In (Adzkiya et al. 2015a., Sec. 5.1), some experiments were carried out in order to test the efficiency of the approach: for any given n it was generated an $n \times n$ matrix A with 2 non-ε elements randomly placed in each row. The finite elements were randomly generated integers between 1 and 100. They claim that the test over a number of randomly generated dynamics goes against biasing the experimental outcomes and allows claiming the applicability of the technique over general MPL systems. Over 10 experiments, for $n = 10$, the average number of regions was 700.80 [regions] and the average time to generate the PWA system was 4.73 [sec]. Note that in this case the worst-case cardinality for the number of regions is $\prod_{i=1}^{10} 2 = 2^{10} = 1024$, since there are only 2 non-ε elements in each row. The experiments were run in a 12-core Intel Xeon 3.47 GHz PC with 24 GB of memory.*

In (ADZKIYA *et al.*, 2015, Sec. 2.2) it is proposed a backtracking technique to improve the performance of Algorithm 2.3. The technique is based on the partial coefficients $(g_1, ..., g_k)$ for $k \in \{1, ..., n\}$, and corresponding region given by:

$$R_{(g_1,...,g_k)} = \bigcap_{i=1}^{k} \bigcap_{\substack{j=1 \\ j \neq g_i}}^{p} \left\{ \mathbf{x} \in \overline{\mathbb{R}}^n_{max} : x_j - x_{g_i} \leq a_{ig_i} - a_{ij} \right\}. \tag{2.58}$$

Note that, for $k > 1$, the partial regions (2.58) can be recursively constructed as:

$$R_{(g_1,...,g_k)} = R_{(g_1,...,g_{k-1})} \bigcap_{\substack{j=1 \\ j \neq g_i}}^{p} \left\{ \mathbf{x} \in \overline{\mathbb{R}}^n_{max} : x_j - x_{g_k} \leq a_{kg_k} - a_{kj} \right\}. \tag{2.59}$$

Thus if the region associated with some partial coefficient $(g_1^\emptyset, ..., g_k^\emptyset)$ is empty, then, for all coefficients $(g_1, ..., g_n)$ such that $g_i = g_i^\emptyset$ for all $i \in \{1, ..., k\}$, the corresponding regions are also empty. Therefore, the computations associated to these coefficients can be skipped, which improves the performance of Algorithm.

Example 2.31 *Given the MPL system described by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 2 & 1 & 4 \\ 5 & 2 & 3 \\ 4 & 3 & 1 \end{pmatrix} \otimes \boldsymbol{x}(k-1),$$

one can verify that the regions associated to the partial coefficients $\boldsymbol{g}^\emptyset \in \{(1,2), (1,3), (2,3)\}$ *are empty. Thus, for all coefficients* $\boldsymbol{g} \in \{(1,2,g_3^\emptyset), (1,3,g_3^\emptyset), (2,3,g_3^\emptyset)\}$*, where* $g_3^\emptyset \in \{1,2,3\}$*, the corresponding region is also empty and the computations associated to these coefficients can be skipped. Indeed, the coefficients with corresponding nonempty region are given by* $\boldsymbol{g} \in \{(1,1,1), (1,1,2), (2,1,1), (2,1,2), (2,2,2), (3,1,2), (3,2,2), (3,3,1), (3,3,2), (3,3,3)\}$.

Given a matrix $A \in \overline{\mathbb{R}}^{n \times p}_{max}$, Algorithm 2.4 describes a general procedure to generate the corresponding PWA system using this backtrack technique.

Algorithm 2.4 works as follows. In setp 1 the output variables are initialized. In step 5 the regions $R_{(i)}$, $i \in \{1, ..., p\}$ are computed. If R_i is not empty the procedure saves the partial coefficient i to the variable G_1 (step 9). In step 17, the partial regions $R_{(g_1,...,g_{k-1},i)}$, $i \in \{1, ..., p\}$ are recursively computed according to (2.59). If $R_{(g_1,...,g_{k-1},i)}$ is not empty the procedure saves coefficient $(g_1, ..., g_{k-1}, i)$ to the variable G_k (step 9). Note that, if the region associated to the partial coefficient $(g_1, ..., g_{k-1}, i)$ is empty, then the coefficient is skipped in the next recursive steps. The affine dynamics (equations (2.56) and (2.57)) are computed in steps 7 and 19. In the last recursive step ($k = n$, step 22) the procedure saves the nonempty regions and corresponding dynamics to the output variables (step 23).

Algorithm 2.4: Expressing an MPL system as a PWA system. The assignment $zeros(.,.)$ generates a matrix of specified dimensions, with entries equal to 0.

input : $A \in \overline{\mathbb{R}}_{max}^{n\times p}$
output: R, A, f

$\mathtt{R} \leftarrow \emptyset$, $\mathtt{A} \leftarrow \emptyset$, $\mathtt{f} \leftarrow \emptyset$, $G_i|_{i=1}^{n} \leftarrow \emptyset$;
for all $\boldsymbol{i} \in (1, ..., p)$ **do**
  $R_{(i)} \leftarrow \mathbb{R}^p$, $A_{(i)} \leftarrow zeros(n,p)$, $f_{(i)} \leftarrow zeros(n,1)$;
  for all $\boldsymbol{j} \in \{1, ..., p\}$ **do**
    $R_{(i)} \leftarrow R_{(i)} \cap \{\mathbf{x} \in \mathbb{R}^p : x_j - x_i \leq a_{1i} - a_{1j}\}$;
  end for
  $A_{(i)}(1,i) = 1$, $f_{(i)}(1) = a_{1i}$;
  if R_i *is not empty* **then**
    $G_1 \leftarrow G_1 \cup \{i\}$;
  end if
end for
for all $k \in \{2, ..., n\}$ **do**
  for all $g = (g_1, ..., g_{k-1}) \in G_{k-1}$ **do**
    for all $\boldsymbol{i} \in \{1, ..., p\}$ **do**
      $R_{(g_1,...,g_{k-1},i)} \leftarrow R_{(g_1,...,g_{k-1})}$, $A_{(g_1,...,g_{k-1},i)} \leftarrow A_{(g_1,...,g_{k-1})}$,
      $f_{(g_1,...,g_{k-1},i)} \leftarrow f_{(g_1,...,g_{k-1})}$;
      for all $\boldsymbol{j} \in \{1, ..., p\}$ **do**
        $R_{(g_1,...,g_{k-1},i)} \leftarrow R_{(g_1,...,g_{k-1},i)} \cap \{\mathbf{x} \in \mathbb{R}^p : x_j - x_i \leq a_{ki} - a_{kj}\}$;
      end for
      $A_{(g_1,...,g_{k-1},i)}(k,i) = 1$, $f_{(g_1,...,g_{k-1},i)}(k) = a_{ki}$;
      if $R_{(g_1,...,g_{k-1},i)}$ *is not empty* **then**
        $G_k \leftarrow G_k \cup \{(g_1, ..., g_{k-1}, i)\}$;
        if $k == n$ **then**
          $\mathtt{R} \leftarrow \mathtt{R} \cup \{R_{(g_1,...,g_{k-1},i)}\}$, $\mathtt{A} \leftarrow \mathtt{A} \cup \{A_{(g_1,...,g_{k-1},i)}\}$,
          $\mathtt{f} \leftarrow \mathtt{f} \cup \{f_{(g_1,...,g_{k-1},i)}\}$;
        end if
      end if
    end for
  end for
end for

2.5.1 DBM Representation of PWA Systems

In this section, the DBM data structure is used to represent PWA systems generated by MPL systems. It is recalled that each component of the PWA system can be represented by a DBM.

As presented in section 2.4 the DBM can represent intersections of finitely many linear inequalities. Thus, in order to represent the PWA systems as DBM, each component of the

PWA system must be expressed as an intersection of linear inequalities. From (2.54) each region $R_{\mathbf{g}}$ is an intersection of linear inequalities. Furthermore, the affine dynamics (2.55) can be expressed as:

$$\bigcap_{i=1}^{p} \{z_i(k) - x_{g_i}(k-1) \leq a_{ig_i}\} \cap \bigcap_{i=1}^{p} \{x_{g_i}(k-1) - z_i(k) \leq -a_{ig_i}\}\,. \tag{2.60}$$

Therefore, each component of the PWA system can be represented by an $(n+p+1) \times (n+p+1)$ DBM, noted by $D^{(\mathbf{g})}$, which constrains the variables $\mathbf{z}(k) = (z_1(k) \ \cdots \ z_n(k))^T$ and $\mathbf{x}(k-1) = (x_1(k-1) \ \cdots \ x_p(k-1))^T$ and their differences. Algorithm 2.5 generates MPL systems as PWA systems using DBM as data structure. The output of the algorithm is a collection of DBM given in the variable `D`. It should be noted that this algorithm is based on the procedure given in Algorithm 2.3, and therefore has the same worst-case cardinality, i.e, $\mathcal{O}\left(p^n(np+p^3)\right)$. Moreover, the backtracking technique presented in Algorithm 2.4 can be used in order to improve the performance of the algorithm.

Example 2.32 *Consider the autonomous MPL system given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 8 & 5 \\ 4 & 3 \end{pmatrix} \otimes \boldsymbol{x}(k-1).$$

Using A as the input of Algorithm 2.5, the output is the collection of DBM `D` $= \{D^{(1,1)}$, $D^{(1,2)}$, $D^{(2,2)}\}$, *where*[3]*:*

$$D^{(1,1)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (8,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (4,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-8,\leq) & (-4,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

$$D^{(1,2)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (8,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (3,\leq) \\ \varepsilon_{\mathcal{B}} & (-8,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-3,\leq) & (3,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

[3] Notation: $\mathbf{x}' \equiv \mathbf{x}(k)$ and $\mathbf{x} \equiv \mathbf{x}(k-1)$

Algorithm 2.5: Expressing an MPL system as a PWA system using DBM as data structure. The assignment $dbmEye(\cdot)$ generates a square matrix of specified dimension, with entries $d_{ij} = e_{\mathcal{B}}$ if $i = j$ and $d_{ij} = \varepsilon_{\mathcal{B}}$ if $i \neq j$. The assignment $dbmNull(\cdot, \cdot)$ generates a matrix of specified dimension, with entries $d_{ij} = \varepsilon_{\mathcal{B}}$.

input : $A \in \overline{\mathbb{R}}^{n\times p}_{max}$
output: `D // A collection of DBM representing the PWA system;`

1 `D` $\leftarrow \emptyset$;
2 **for all** $\boldsymbol{g} \in \{1, ..., p\}^n$ **do**
3 $\quad R_{\mathbf{g}} \leftarrow dbmEye(n)$, $dynSup \leftarrow dbmNull(n, p)$, $dynInf \leftarrow dbmNull(p, n)$;
4 $\quad$ **for all** $i \in \{1, ..., n\}$ **do**
5 $\quad\quad$ **if** $a_{i,g_i} \neq \epsilon$ **then**
6 $\quad\quad\quad$ **for all** $j \in \{1, ..., n\}$ **do**
7 $\quad\quad\quad\quad$ **if** $a_{i,j} \neq \epsilon$ **then**
8 $\quad\quad\quad\quad\quad R_{\mathbf{g}}[i, g_i] \leftarrow (\min\{R_{\mathbf{g}}[i, g_i], a_{i,g_i} - a_{i,j}\}, \leq)$ `// define` $R_{\mathbf{g}}$ `(see` (2.54)`)`
9 $\quad\quad\quad\quad$ **end if**
10 $\quad\quad\quad$ **end for**
11 $\quad\quad\quad dynSup[i, g_i] \leftarrow (a_{i,g_i}, \leq)$ `// represents` $z_i \leq x_{g_i} + a_{i,g_i}$
12 $\quad\quad\quad dynInf[g_i, i] \leftarrow (-a_{i,g_i}, \leq)$ `// represents` $z_i \geq x_{g_i} + a_{i,g_i}$
13 $\quad\quad$ **end if**
14 $\quad$ **end for**
15 $\quad$ **if** $R_{\boldsymbol{g}}$ *is not empty* **then**
16 $\quad\quad D^{(\mathbf{g})} \leftarrow dbmEye(n + p + 1)$ `//`
17 $\quad\quad D^{(\mathbf{g})}[2 : n + 1,\ n + 2 : n + p + 1] \leftarrow dynSup$ `//`
18 $\quad\quad D^{(\mathbf{g})}[n + 2 : n + p + 1,\ 2 : n + 1] \leftarrow dynInf$ `//`
19 $\quad\quad D^{(\mathbf{g})}[n + 2 : n + p + 1,\ n + 2 : n + p + 1] \leftarrow R_{\mathbf{g}}$ `//`
20 $\quad\quad$ `D` $\leftarrow$ `D` $\cup \{D^{(\mathbf{g})}\}$;
21 $\quad$ **end if**
22 **end for**

$$D^{(\mathbf{g})} = \begin{array}{c} \begin{array}{ccc} x_0 & z_1 \ldots z_n & x_1 \ldots x_p \end{array} \\ \left(\begin{array}{c|c|c} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \ldots \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \ldots \varepsilon_{\mathcal{B}} \\ \hline \varepsilon_{\mathcal{B}} \\ \vdots & e_{\mathcal{B}}^{n\times n} & \begin{array}{c} dunSup \\ \text{(step 17)} \end{array} \\ \varepsilon_{\mathcal{B}} \\ \hline \varepsilon_{\mathcal{B}} \\ \vdots & \begin{array}{c} dunInf \\ \text{(step 18)} \end{array} & \begin{array}{c} R_{\mathbf{g}} \\ \text{(step 19)} \end{array} \\ \varepsilon_{\mathcal{B}} \end{array}\right) \begin{array}{c} x_0 \\ z_1 \\ \vdots \\ z_n \\ x_1 \\ \vdots \\ x_p \end{array} \end{array}$$

$$D^{(2,2)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5, \leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (3, \leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-3, \leq) \\ \varepsilon_{\mathcal{B}} & (-5, \leq) & (-3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array} \end{array}$$

Example 2.33 *Consider the nonautonomous MPL system of example 2.29:*

$$\boldsymbol{x}' = \begin{pmatrix} 2 & 4 \\ 3 & e \end{pmatrix} \otimes \boldsymbol{x} \oplus \begin{pmatrix} e \\ \varepsilon \end{pmatrix} \otimes \boldsymbol{u}.$$

The corresponding augmented autonomous MPL system is characterized by the matrix:

$$\bar{A} = \begin{pmatrix} 2 & 4 & e \\ 3 & e & \varepsilon \end{pmatrix}.$$

Thus, using $\bar{A}$ as the input of Algorithm 2.5, the output is the collection of DBM $\mathtt{D} = \{D^{(1,1)},\, D^{(2,1)},\, D^{(2,2)},\, D^{(3,1)},\, D^{(3,2)}\}$, *where*[4]*:*

$$D^{(1,1)} = \begin{array}{cc} \begin{matrix} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{matrix} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (3,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-2,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} & \begin{matrix} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{matrix} \end{array}$$

$$D^{(2,1)} = \begin{array}{cc} \begin{matrix} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{matrix} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (3,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-3,\leq) & e_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{matrix} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{matrix} \end{array}$$

$$D^{(2,2)} = \begin{array}{cc} \begin{matrix} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{matrix} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-3,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{matrix} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{matrix} \end{array}$$

$$D^{(3,1)} = \begin{array}{cc} \begin{matrix} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{matrix} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (3,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-2,\leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} & (-4,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} & \begin{matrix} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{matrix} \end{array}$$

[4] Notation: $\mathbf{x}' \equiv \mathbf{x}(k)$, $\mathbf{x} \equiv \mathbf{x}(k-1)$ and $\mathbf{u} \equiv \mathbf{u}(k)$

$$D^{(3,2)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{array} \\ \left(\begin{array}{cccccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-3, \leq) & (-2, \leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-4, \leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{array}$$

3 Reachability Analysis of MPL Systems

This chapter summarizes the results on reachability analysis of MPL systems presented in (ADZKIYA *et al.*, 2014b; ADZKIYA *et al.*, 2014a; ADZKIYA *et al.*, 2015). It will be shown that it is possible to map DBM-sets through MPL systems. Then, forward and backward reachability analysis will be introduced.

Proposition 3.1 is the basis for the reachability analysis of MPL systems using the PWA-DBM approach.

Proposition 3.1 (ADZKIYA *et al.*, 2015, Th. 1) *The image and the inverse image of a set represented by a DBM w.r.t. a subsystem of a PWA system generated by an MPL system is a set that can be represented by a DBM.*

Proof:

The proof will be given for the image instance. The proof for the inverse image is similar. Each subsystem of a PWA system can be represented by[1]*:*

$$x_i(k) = a_{ig_i} + x_{g_i}(k-1) \text{ if } \boldsymbol{x}(k-1) \in R_{\boldsymbol{g}},\ \forall i \in \{1, ..., n\} \cup \{0\},$$

where, for all $\boldsymbol{g}$*,* g_0 *is set to* 0*,* $a_{00} = 0$*,* $a_{0j} = \varepsilon$ *for all* $j \in \{1, ..., p\}$ *and* $a_{i0} = \varepsilon$ *for all* $i \in \{1, ..., n\}$*.*

Note that, given a set X_{k-1}*, only the points in the intersection* $X_{k-1} \cap R_{\boldsymbol{g}}$ *are governed by this dynamics i.e.:*

$$x_i(k) = a_{ig_i} + x_{g_i}(k-1) \text{ if } \boldsymbol{x}(k-1) \in X_{k-1} \cap R_{\boldsymbol{g}},\ \forall i. \tag{3.1}$$

If X_{k-1} *can be represented by a DBM, the intersection* $X_{k-1} \cap R_{\boldsymbol{g}}$ *can also be represented by a DBM that will be noted by* $D^{(X_{k-1} \cap R_{\boldsymbol{g}})}$*, with entries* $d_{ij}^{(X_{k-1} \cap R_{\boldsymbol{g}})} = (\mathtt{d}_{ij}^{(X_{k-1} \cap R_{\boldsymbol{g}})}, \leq)$*. Since computing the canonical form does not change the region represented by a DBM, it will be assumed that* $D^{(X_{k-1} \cap R_{\boldsymbol{g}}^u)}$ *is in the canonical form. Therefore, for all* $\boldsymbol{x}(k-1) \in X_{k-1} \cap R_{\boldsymbol{g}}$ *we have that the tightest possible upper bound for* $x_i(k-1) - x_j(k-1)$ *is given by:*

$$x_i(k-1) - x_j(k-1) \leq \mathtt{d}_{ij}^{(X_{k-1} \cap R_{\boldsymbol{g}})},\ \forall i, j.$$

[1] This model considers an additional equation corresponding to the artificial variable: $x_0 = 0 + x_0$

In particular:

$$x_{g_i}(k-1) - x_{g_j}(k-1) \leq \mathtt{d}_{g_i g_j}^{(X_{k-1} \cap R_g)}, \ \forall i, j.$$

Adding $a_{ig_i} - a_{jg_j}$ in both sides of the inequality one obtains:

$$\overbrace{a_{ig_i} + x_{g_i}(k-1)}^{x_i(k)} - (\overbrace{a_{jg_j} + x_{g_j}(k-1)}^{x_j(k)}) \leq \mathtt{d}_{g_i g_j}^{(X_{k-1} \cap R_g)} + a_{ig_i} - a_{jg_j}, \ \forall i, j.$$

Thus, the tightest possible upper bound for $x_i(k) - x_j(k)$ is given by:

$$x_i(k) - x_j(k) \leq \mathtt{d}_{g_i g_j}^{(X_{k-1} \cap R_g)} + a_{ig_i} - a_{jg_j}, \ \forall i, j. \tag{3.2}$$

It should be noted that all points in the image of X_{k-1} w.r.t. the subsystem $\boldsymbol{g}$ of the PWA system must satisfy (3.2)*. Otherwise, at least one of the restrictions defined by the dynamics* (3.1) *would be violated. Moreover, all the points that satisfy* (3.2) *can be reached from $X_{k-1} \cap R_{\boldsymbol{g}}$. Therefore the image of X_{k-1} w.r.t. the subsystem $\boldsymbol{g}$ of a PWA system is given by the region defined by* (3.2)*, which can be represented by a DBM $D^{(X_k|g)}$ with entries:*

$$d_{ij}^{(X_k|g)} = (\mathtt{d}_{g_i g_j}^{(X_{k-1} \cap R_g)} + a_{ig_i} - a_{jg_j}, \ \leq). \tag{3.3}$$

■

Given a DBM $D^{(X_{k-1})}$ representing a set X_{k-1}, Algorithm 3.1 computes the image of X_{k-1} w.r.t. a subsystem of a PWA system generated by an MPL system.

In the following is a discussion on how Algorithm 3.1 yields the region defined (3.3), which represents the image of a set X_{k-1} w.r.t. a subsystem $\mathbf{g}$ of the PWA system. Note that, the DBM $D^{(\bar{X}_k)}$ obtained in step 3 of algorithm 3.1 exactly represents (3.1). Moreover, by definition, the DBM obtained in step 4 (which is the canonical form representation of $D^{(\bar{X}_k)}$) has the tightest possible bounds. Therefore, the DBM $D^{(X_k|\mathbf{g})}$, obtained in the step 5 as orthogonal projection of the canonical form over the variables $\mathbf{x}(k)$, is the DBM defined by (3.3).

Similarly, given a DBM $D^{(X_{-k+1})}$ representing a set X_{-k+1}, Algorithm 3.2 computes the inverse image of X_{-k+1} w.r.t. a subsystem of a PWA system generated by an MPL system.

The worst-case complexity of Algorithms 3.1 and 3.2 critically depends on computing the canonical form representation of a DBM in $\mathcal{B}^{(n+p+1)\times(n+p+1)}$ (step 4 for both algorithms), which has cubic complexity w.r.t its dimensions. Thus, the worst-case complexity is $\mathcal{O}((n+p)^3)$ (ADZKIYA *et al.*, 2015, Sec. 2.3).

Algorithm 3.1: Computing the image of a DBM w.r.t a PWA system generated by an MPL system

input : $D^{(X_{k-1})} \in \mathcal{B}^{(p+1)\times(p+1)}$ `// a DBM representing a region` $X_{k-1} \in \mathbb{R}^p$.

: $D^{(\mathbf{g})} \in \mathcal{B}^{(n+p+1)\times(n+p+1)}$ `// a DBM representing a subsystem of a PWA system generated by a matrix` $A \in \overline{\mathbb{R}}^{n\times p}_{max}$.

output: $D^{(X_k|\mathbf{g})} \in \mathcal{B}^{(n+1)\times(n+1)}$ `// a DBM representing the image of` X_{k-1} `w.r.t. the subsystem` **g** `of the PWA system.`

1 $D^{(\mathbb{R}^n)} \leftarrow e_{\mathcal{B}^{n+1\times n+1}}$ `// a DBM representing` $\mathbb{R}^n$

2 $D^{(\mathbb{R}^n \times X_{k-1})} \leftarrow D^{(\mathbb{R}^n)} \times D^{(X_{k-1})}$ `// compute the cart. product (see section 2.4.2)`

3 $D^{(\bar{X}_k)} \leftarrow D^{(\mathbb{R}^n \times X_{k-1})} \oplus_{\mathcal{B}} D^{\mathbf{g}}$ `// compute the intersection (see remark 2.21).`

4 $D^{(\bar{X}_k)} \leftarrow cf(D^{(\bar{X}_k)})$ `// compute the canonical form (see section 2.4.1).`

5 $D^{(X_k|\mathbf{g})} \leftarrow D^{(\bar{X}_k)}\lceil_{x'_1,\ldots,x'_n}$ `// compute the orthogonal projection over` $\mathbf{x}(k)$ `(see section 2.4.2).`

Algorithm 3.2: Computing the inverse image of a DBM w.r.t a PWA system generated by an MPL system

input : $D^{(X_{-k+1})} \in \mathcal{B}^{(n+1)\times(n+1)}$ `// a DBM representing a region` $X_{-k+1} \in \mathbb{R}^n$.

: $D^{(\mathbf{g})} \in \mathcal{B}^{(n+p+1)\times(n+p+1)}$ `// a DBM representing a subsystem of a PWA system generated by a matrix` $A \in \overline{\mathbb{R}}^{n\times p}_{max}$.

output: $D^{(X_{-k}|\mathbf{g})} \in \mathcal{B}^{(p+1)\times(p+1)}$ `// a DBM representing the inverse image of` X_{-k+1} `w.r.t. the subsystem` **g** `of the PWA system.`

1 $D^{(\mathbb{R}^p)} \leftarrow e_{\mathcal{B}^{p+1\times p+1}}$ `// a DBM representing` $\mathbb{R}^n$

2 $D^{(X_{-k+1} \times \mathbb{R}^p)} \leftarrow D^{(X_{-k+1})} \times D^{(\mathbb{R}^n)}$ `// compute the cart. product (see section 2.4.2)`

3 $D^{(\bar{X}_{-k})} \leftarrow D^{(X_{-k+1} \times \mathbb{R}^p)} \oplus_{\mathcal{B}} D^{\mathbf{g}}$ `// compute the intersection (see remark 2.21).`

4 $D^{(\bar{X}_{-k})} \leftarrow cf(D^{(\bar{X}_{-k})})$ `// compute the canonical form (see section 2.4.1).`

5 $D^{(X_{-k}|\mathbf{g})} \leftarrow D^{(\bar{X}_{-k})}\lceil_{x_1,\ldots,x_n}$ `// compute the orthogonal projection over` $\mathbf{x}(k-1)$ `(see section 2.4.2).`

Corollary 3.2 (ADZKIYA *et al.*, 2015, Cor. 5) *The image of a union of finitely many DBM w.r.t. a PWA system generated by an MPL model is a union of finitely many DBM.*

Given a PWA system generated by a matrix $A \in \overline{\mathbb{R}}^{n\times p}_{max}$, computing the image (or the inverse image) of a union of q DBM can be done by computing the image (or the inverse image) of each DBM w.r.t each subsystem of the PWA system. Thus the worst-case complexity depends on the number of DBM (considered to be q), on the worst-case cardinality of the collection of subsystem, given by p^n and on the complexity of computing the image (or the inverse image) of each DBM w.r.t. each subsystem of a PWA system, which is $\mathcal{O}((n+p)^3)$.

Therefore, the worst-case complexity is $\mathcal{O}(qp^n(n+p)^3)$ (ADZKIYA *et al.*, 2015, Sec. 2.3).

Remark 3.3 *For autonomous MPL systems, parameter p equals n, and therefore the worst-case complexity of computing the image (or the inverse image) of q DBM w.r.t the system is $\mathcal{O}(qn^{n+3})$. For nonautonomous MPL systems, parameter p equals $n+m$, and therefore the worst-case complexity is $\mathcal{O}(q(n+m)^{n+3})$.*

Sections 3.1 and 3.2 introduce forward and backward reachability analysis, respectively. It will be assumed that the set of initial/final conditions $X_0 \subseteq \mathbb{R}^n$ and the set of control at each event step $U_k \subseteq \mathbb{R}^m$ can be represented by a union of q_0 and r_k DBM, respectively. Moreover, the cardinality of the DBM union set representing X_k at event step k will be noted by q_k.

3.1 Forward Reachability Analysis

The forward reachability analysis of MPL systems concerns the computation of the set of all states that can be reached from a set of initial states via MPL dynamics, at a particular event step (the reach set) or over a set of consecutive events (reach tube). Formal definitions of reach sets and reach tube are given in the following.

Definition 3.4 (reach set (ADZKIYA *et al.*, 2014b, Def. 3)) *Given an MPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach set** X_N at the event step $N > 0$ is the set of all states $\{\boldsymbol{x}(N) : \boldsymbol{x}(0) \in X_0\}$ obtained via the MPL dynamics, possibly by application of controls.*

Definition 3.5 (reach tube (ADZKIYA *et al.*, 2014b, Def. 4)) *Given an MPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach tube** is defined by the set-valued function $k \mapsto X_k$ for any given $k > 0$ where X_k is defined.*

Given a set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the reach tube can be computed by using the one-step dynamics for autonomous and nonautonomous MPL systems iteratively: at each event step, the PWA system (and corresponding DBM representation) generated by the MPL system is used to compute the successive reach set. Section 3.1.1 presents a procedure to compute, recursively, the reach tube with focus on *autonomous* MPL systems (ADZKIYA *et al.*, 2014b) and section 3.1.2 presents a generalization of the approach to *nonautonomous* MPL systems (ADZKIYA *et al.*, 2015).

3.1.1 Forward Reachability Analysis of Autonomous MPL systems

Given an *autonomous* MPL system and a nonempty set of initial conditions X_0, the reach set X_k at the event step k can be recursively calculated as the image of the reach set X_{k-1} w.r.t the MPL dynamics:

$$X_k = \mathcal{I}_A\{X_{k-1}\} = \{A \otimes \mathbf{x} : \mathbf{x} \in X_{k-1}\}. \tag{3.4}$$

From Corollary 3.2, if X_{k-1} can be represented by a union of q_{k-1} DBM, then $X_k = \mathcal{I}_A\{X_{k-1}\}$ can be represented by a union of q_k DBM. Thus, by induction, it can be concluded that if X_0 can be represented by a union of q_0 DBM, then X_k can be represented by a union of q_k DBM, for each $k \in \mathbb{N}$.

Given the set of initial conditions X_0, computing the reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct PWA system generated by $A \in \overline{\mathbb{R}}_{max}^{n\times n}$; then, for each k, compute the image of X_{k-1} w.r.t. the PWA system. The reach tube is then obtained by aggregating the reach sets. The worst-case complexity to characterize MPL systems via PWA dynamics is $\mathcal{O}(n^{n+3})$ (see algorithm 2.5). Furthermore, the worst-case complexity to compute $\mathcal{I}_A\{X_{k-1}\}$, for each k is $\mathcal{O}(q_{k-1}n^{n+3})$ (see remark 3.3). Thus, the overall complexity to compute the reach tube is $\mathcal{O}(n^{n+3}\sum_{k=1}^{N} q_{k-1})$.

Remark 3.6 *Given the cardinality q_{k-1} of the DBM union set at event step $k-1$, the worst-case cardinality q_k is $q_{k-1}n^n$, which corresponds to the maximum possible number of nonempty DBM representing the image of the intersection of each DBM at $k-1$ and each region of the partitioned system. In practice, many regions are empty, and even for nonempty regions, many intersections of DBM and regions are also empty, then the cardinality q_k is drastically smaller than its worst-case bound. However, in general, it is not possible to quantify the exact cardinality q_k a priori (ADZKIYA* et al.*, 2015, Sec. 5).*

In general, in order to compute X_N, it is necessary to compute $X_1, ..., X_{N-1}$. However, there are cases in which the structure of the MPL dynamics leads to savings for the computation of the reach tube. Consider the case in which the state matrix of an autonomous MPL system is irreducible. According to corollary 2.14 there exists $k_0(X_0) = \max_{\mathbf{x}\in X_0} k_0(\mathbf{x})$ such that, for all $k \geq k_0(X_0)$, $X_{k+c} = \lambda^c \otimes X_k$, where c is the cyclicity of the critical graph of A and λ is the max-plus eigenvalue of A. In this case, in order do compute X_N, $N > k_0(X_0)$, it is only necessary to compute $X_1, ..., X_{k_0(X_0)}$. Furthermore, if X_0 can be represented by a union of finitely many stripes[2], the infinite-horizon reach tube is also a union of finitely many stripes and can be computed explicitly in finite time (ADZKIYA *et al.*, 2014b, Th. 1). The

[2] A stripe is an unbounded region, i.e., for each j, $-\infty \leq x_j \leq \infty$ (see definition 2.24).

claim follows by noticing that the image of a union of finitely many stripes w.r.t. a PWA system generated by an MPL model is a union of finitely many stripes. Then, since a stripe is a collection of equivalence classes (HEIDERGOTT *et al.*, 2006, Sec 1.4), $\alpha \otimes X_k = X_k$ for all $\alpha \in \mathbb{R}$. Thus it follows from corollary 2.14 that $X_{k+c} = X_k$ for all $k \geq k_0(X_0)$, and therefore the infinite-horizon reach tube is $\bigcup_{i=0}^{\infty} X_i = \bigcup_{i=0}^{k_0(X_0)+c-1} X_i$.

The reach set for a specific event step N can be computed using a one-shot procedure. Given a nonempty set of initial conditions X_0, the reach set X_N at the event step N is given by:

$$X_N = \mathcal{I}_{A^{\otimes N}}\{X_0\} = \{A^{\otimes N} \otimes \mathbf{x} : \mathbf{x} \in X_0\}. \tag{3.5}$$

A general procedure for computing X_N is: 1) compute $A^{\otimes N}$; then, 2) construct the PWA system generated by $A^{\otimes N}$; and, 3) compute the image of X_0 w.r.t. the obtained PWA system. The overall complexity of this procedure is $\mathcal{O}([\log_2(N)]n^3 + q_0 N^3)$ (see, (ADZKIYA *et al.*, 2014b, Sec. 3.2)), where q_0 is the cardinality of the DBM union set representing X_0.

3.1.2 Forward Reachability Analysis of Nonautonomous MPL systems

A similar procedure for forward reachability analysis of nonautonomous MPL systems can be defined. First, the nonautonomous MPL system is represented as an augmented autonomous MPL system (see equation 2.37); then, given a nonempty set of initial conditions X_0 and the set of inputs U_k for $k \in \{1, ..., N\}$, the reach set X_k at the event step k can be recursively calculated as the image of $X_{k-1} \times U_k$ w.r.t the augmented MPL system:

$$X_k = \mathcal{I}_F\{X_{k-1} \times U_k\} = \{F \otimes \mathbf{y} : \mathbf{y} \in X_{k-1} \times U_k\}. \tag{3.6}$$

If X_{k-1} can be represented by a union of q_{k-1} DBM and U_k can be represented by a union of r_k DBM, then $X_{k-1} \times U_k$ can be represented by a union of $\bar{q}_{k-1} = q_{k-1} r_k$ DBM. Thus, from Corollary 3.2, $X_k = \mathcal{I}_F\{X_{k-1} \times U_k\}$ can be represented by a union of q_k DBM. By induction, it can be concluded that if X_0 can be represented by a union of q_0 DBM and U_k can be represented by a union of r_k DBM for each $k \in \mathbb{N}$, then X_k can be represented by a union of q_k DBM, for each $k \in \mathbb{N}$.

Given a nonautonomous MPL system, the set of initial conditions X_0 and set of inputs U_k for each $k \in \{1, ..., N\}$, computing the reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the PWA system generated by $F \in \overline{\mathbb{R}}_{max}^{n \times (n+m)}$; then, for each $k \in \{1, ..., N\}$, compute the image of $X_{k-1} \times U_k$ w.r.t. PWA system. The worst-case complexity to characterize the MPL system via PWA dynamics is $\mathcal{O}((n+m)^{n+3}$ (see algorithm 2.5).

Furthermore, the worst-case complexity to compute $\mathcal{I}_F\{X_{k-1} \times U_k\}$, for each $k \in \{1, ..., N\}$ is $\mathcal{O}(\bar{q}_{k-1}(n+m)^{n+3})$ (see remark 3.3). Thus, the overall complexity to compute the reach tube is $\mathcal{O}((n+m)^{n+3}\sum_{k=1}^{N}\bar{q}_{k-1})$.

For nonautonomous MPL systems, the reach set for a specific event step N can also be computed using a one-shot procedure. Given a nonempty set of initial conditions X_0, the reach set X_N at the event step N is given by:

$$X_N = (A^{\otimes N},\ A^{\otimes (N-1)} \otimes B, ..., B) \otimes (X_0 \times U_1 \times ... \times U_N). \tag{3.7}$$

Given the matrices $A \in \overline{\mathbb{R}}^{n\times n}_{max}$ and $B \in \overline{\mathbb{R}}^{n\times m}_{max}$, a set of initial conditions X_0 (represented by a union of q_0 DBM) and a sequence of input sets $U_1, ..., U_N$, a general procedure for computing X_N is given by: 1) generate the matrix $(A^{\otimes N},\ A^{\otimes (N-1)} \otimes B, ..., B)$; then, 2)Construct the PWA system generated by this matrix; and, 3) compute the image of $X_0 \times U_1 \times ... \times U_N$ w.r.t the obtained PWA system. The complexity of steps 1, 2 and 3 is, respectively, $\mathcal{O}(Nn^3 + Nn^2m)$, $\mathcal{O}((n+mN)^{n+3})$ and $\mathcal{O}(q_0(n+mN)^{n+3})$. Note that , this approach is not tractable for problems over long event horizons, since the maximum number of regions of the PWA system is $(n+mN)^n$ and grows polynomially w.r.t. the event horizon N (ADZKIYA *et al.*, 2015, Sec. 3.2).

3.2 Backward Reachability Analysis

The backward reachability analysis of MPL systems concerns the computation of the set of all states that leads to a set of initial states via MPL dynamics, at a particular event step (backward reach set) or over a set of consecutive events (backward reach tube).

Definition 3.7 (backward reach set (ADZKIYA *et al.*, 2014a, Def. 7)) *Given an MPL system and a nonempty set of final positions $X_0 \subseteq \mathbb{R}^n$, the **backward reach set** X_{-N} is the set of all states $\boldsymbol{x}(-N)$ that leads to X_0 in N steps of the MPL dynamics, possibly by application of controls.*

Definition 3.8 (backward reach tube (ADZKIYA *et al.*, 2014a, Def. 8)) *Given an MPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach tube** is defined by the set-valued function $k \mapsto X_{-k}$ for any given $k > 0$ where X_{-k} is defined.*

Similar to the forward reachability instance, given a set of final conditions $X_0 \subseteq \mathbb{R}^n$, the reach tube can be computed by using the one-step dynamics for autonomous and nonautonomous MPL systems iteratively. Section 3.2.1 presents procedure to compute, recursively, the backward reach tube with focus on *autonomous* MPL systems (ADZKIYA *et al.*, 2014a) and section 3.2.2 presents an generalization of the approach to *nonautonomous* MPL systems (ADZKIYA *et al.*, 2015).

3.2.1 Backward Reachability Analysis of Autonomous MPL systems

Given a *autonomous* MPL system and a nonempty set of final conditions X_0, the backward reach set X_{-k} can be recursively calculated as the inverse image of the reach set X_{-k+1} w.r.t the MPL dynamics:

$$X_{-k} = \mathcal{I}_A^{-1}\{X_{-k+1}\} = \{\mathbf{x} \in \mathbb{R}^n : A \otimes \mathbf{x} \in X_{-k+1}\}. \tag{3.8}$$

From Corollary 3.2 it can be shown that if X_0 can be represented by a union of q_0 DBM, then X_{-k} can be represented by a union of q_{-k} DBM, for each $k \in \mathbb{N}$.

Given the set of final conditions X_0, computing the backward reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the PWA system generated by A; then, for each $k \in \mathbb{N}$, compute the inverse image of X_{k-1} w.r.t. the PWA system. The worst-case complexity to compute $\mathcal{I}_A^{-1}\{X_{-k+1}\}$, for each $k \in \mathbb{N}$ is $\mathcal{O}(q_{-k+1}n^{n+3})$. Thus, the overall complexity is $\mathcal{O}(n^{n+3}\sum_{k=1}^{N} q_{-k+1})$.

Similarly to the forward reachability instance, there are cases in which the infinite-horizon backward reach tube can be explicitly computed. If the MPL system is irreducible and X_0 is not intersected with the complete periodic behavior[3], i.e., $X_0 \cap E(A^{\otimes c}) = \emptyset$, there exists a finite k_ϕ such that X_{-k} is empty for all $k \geq k_\phi$ (ADZKIYA *et al.*, 2014a, Prop. 9). Note that, if $X_0 \cap E(A^{\otimes c}) = \emptyset$, all $\mathbf{x} \in X_0$ belongs to the transient behavior of the system, and therefore the minimum length of the transient part of X_0 is positive, i.e. $min_{\mathbf{x}\in X_0} k_0(\mathbf{x}) > 0$. Furthermore, if the backward reach set X_{-k} is not empty, all $\mathbf{x} \in X_{-k}$ is also in the transient behavior of the system and the minimum length of the transient part of X_{-k} is increasing with k as follows: $min_{\mathbf{x}\in X_{-k}} k_0(\mathbf{x}) = k + min_{\mathbf{x}\in X_0} k_0(\mathbf{x})$. However, the maximum length of the transient part of X_{-k} is bounded by $\min_{\mathbf{x}\in X_{-k}} k_0(\mathbf{x}) \leq \max_{\mathbf{x}\in X_{-k}} k_0(\mathbf{x}) \leq K_0(A)$ (see Remark 2.15) whenever X_{-k} is not empty. Therefore, X_{-k} is empty if $k > K_0(A)$, which would imply $\min_{\mathbf{x}\in X_{-k}} k_0(\mathbf{x}) > K_0(A)$.

The set of all states that can lead to a given set of final positions X_0 in N event steps (i.e., the backward reach set X_{-N}) can be computed using a one-shot procedure. Given a nonempty set of final conditions X_0, the backward reach set X_{-N} is given by:

$$X_{-N} = \mathcal{I}_{A^{\otimes N}}^{-1}\{X_0\} = \{\mathbf{x} \in \mathbb{R}^n : A^{\otimes N} \otimes \mathbf{x} \in X_0\}. \tag{3.9}$$

[3] The complete set of periodic behaviors is given by the eigenspace of $A^{\otimes c}$, i.e. $E(A^{\otimes c})$ where c is the cyclicity of the critical graph of A (recall the definition of eigenspace in Proposition 2.12)

A general procedure for computing X_{-N} is: 1) compute $A^{\otimes N}$; then, 2) construct the PWA system generated by $A^{\otimes N}$; and, 3) compute the inverse image of X_0 w.r.t. the obtained PWA system. the overall complexity of the one-shot computation of the backward reach set is the same as the forward instance for autonomous uMPL systems.

3.2.2 Backward Reachability Analysis of Nonautonomous MPL systems

To proceed with the backward reachability analysis of nonautonomous MPL systems, the system is first represented as an equivalent augmented autonomous MPL system (see equation (2.37)); then, given a set of final conditions X_0 and the set of inputs U_{-k} for each $k \in \mathbb{N}$, the backward reach set X_{-k} can be recursively calculated as the inverse image of X_{-k+1}:

$$X_{-k} = \mathcal{I}_F^{-1}\{X_{-k+1}\} = \{\mathbf{x} \in \mathbb{R}^n : \exists \mathbf{u} \in U_{-k+1} : F \otimes (\mathbf{x}^T\ \mathbf{u}^T)^T \in X_{-k+1}\}. \quad (3.10)$$

Given a nonautonomous MPL system, the set of final conditions X_0 and set of inputs U_{-k} for each $k \in \{0, ..., N-1\}$, computing the backward reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the PWA system generated by $[\mathbf{F}] = ([\mathbf{A}]\ [\mathbf{B}])$; then, for each $k \in \{1, ..., N\}$, compute the inverse image of X_{k-1} w.r.t. the PWA system; next, intersect the inverse image with $\mathbb{R}^n \times U_{-k+1}$; and finally, project the intersection over the state variables.

From Corollary 5.2, it can be shown that X_{-k} can be represented by a union of finitely many DBM. The worst-case complexity to compute $\mathcal{I}_F^{-1}\{X_{-k+1}\}$ is $\mathcal{O}(\bar{q}_{-k+1}(n+m)^{n+3})$, where: $\bar{q}_{-k+1} = q_{-k+1}r_{-k+1}$ and q_{-k+1} and r_{-k+1} are, respectively, the cardinality of the DBM union set representing X_{-k+1} and U_{-k+1}. Thus, the overall complexity to compute X_{-N} is $\mathcal{O}((n+m)^{n+3}\sum_{k=1}^{N}\bar{q}_{-k+1})$.

In the following it is presented a one-shot procedure for computing the backward reach set X_{-N}, for a particular index N. Given a nonempty set of final conditions X_0, the set of all states that are able to enter X_0 in N event steps is given by:

$$\begin{aligned} X_{-N} \quad = \quad & \{\mathbf{x}(-N) \in \mathbb{R}^n : \exists \mathbf{u}(-N+1) \in U_{-N+1}, ..., \mathbf{u}(0) \in U_0 \\ & : (A^{\otimes N},\ A^{\otimes (N-1)} \otimes B, ..., B) \otimes (\mathbf{x}(-N)^T\ \mathbf{u}(-N+1)^T\ \mathbf{u}(0)^T)^T \in X_0\}. \end{aligned} \quad (3.11)$$

Given the matrices $A \in \overline{\mathbb{R}}_{max}^{n \times n}$ and $B \in \overline{\mathbb{R}}_{max}^{n \times m}$, a set of final positions X_0 and a sequence of input sets $U_{-N+1}, ..., U_0$, a general procedure for computing X_{-N} is given by: 1) generate the matrix $[A^{\otimes N},\ A^{\otimes (N-1)} \otimes B, ..., B]$; then, 2) Construct the PWA system generated by this matrix; 3) compute the inverse image of X_0 w.r.t the obtained PWA system; 4) intersect the inverse image with $\mathbb{R}^n \times U_1 \times ... \times U_N$; and finally, 5) project the intersection over the state variables. The complexity of this procedure is the same as the one-shot procedure for the forward case presented in section 3.1.2.

Example 3.9 *Consider the autonomous MPL system given by:*

$$\boldsymbol{x}(k) = \begin{pmatrix} 8 & 5 \\ 4 & 3 \end{pmatrix} \otimes \boldsymbol{x}(k-1).$$

In example 2.32 this system was represented as a collection of DBM $\mathtt{D}^{(PWA)} = \{D^{(1,1)}, D^{(1,2)}, D^{(2,2)}\}$.

Let us now compute the reach sets X_k *for* $k \in \{1,2,3\}$ *and the backward reach sets* X_{-k} *for* $k \in \{1,2\}$ *given* $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \leq x_1 \leq 2,\ -4 \leq x_2 \leq 6\}$. *Note that the set* X_0 *can be represented by the following DBM*[4]*:*

$$D^{(X_0)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (4,\leq) \\ (2,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (6,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

To compute the reach set $X_1 = \mathcal{I}_A\{X_0\}$, *we must compute the image of* X_0 *w.r.t each component* $\boldsymbol{g}$ *of the PWA system. According to algorithm 3.1, the image of* $D^{(X_0)}$ *w.r.t.* $D^{(1,1)}$ *can be computed as follows: first, we compute the Cartesian product of* $D^{(\mathbb{R}^2)}$ *and* $D^{(X_0)}$*:*

$$D^{(\mathbb{R}^2 \times X_0)} = D^{(\mathbb{R}^2)} \times D^{(X_0)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (4,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (6,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array}$$

Then, we compute the intersection of $D^{(\mathbb{R}^2 \times X_0)}$ *and* $D^{(1,1)}$*:*

[4] Notation: $\mathbf{x}' \equiv \mathbf{x}(k)$ and $\mathbf{x} \equiv \mathbf{x}(k-1)$.

$$
D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\
\begin{pmatrix}
e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) \\
\varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (8, \leq) & \varepsilon_{\mathcal{B}} \\
\varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\
(2, \leq) & (-8, \leq) & (-4, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\
(6, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (1, \leq) & e_{\mathcal{B}}
\end{pmatrix}
\end{array}
\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}
$$

Next, we compute the canonical form representation of the intersection:

$$
cf(D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)}) =
\begin{array}{c}
\begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\
\begin{pmatrix}
e_{\mathcal{B}} & (-8, \leq) & (-4, \leq) & e_{\mathcal{B}} & (4, \leq) \\
(10, \leq) & e_{\mathcal{B}} & (4, \leq) & (8, \leq) & (14, \leq) \\
(6, \leq) & (-4, \leq) & e_{\mathcal{B}} & (4, \leq) & (10, \leq) \\
(2, \leq) & (-8, \leq) & (-4, \leq) & e_{\mathcal{B}} & (6, \leq) \\
(3, \leq) & (-7, \leq) & (-3, \leq) & (1, \leq) & e_{\mathcal{B}}
\end{pmatrix}
\end{array}
\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}
$$

Finally, we compute the orthogonal projection of the canonical form over the variables x_1' and x_2':

$$
D^{(X_1|_{\boldsymbol{g}=(1,1)})} = cf(D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)}) \lceil_{\boldsymbol{x}'} =
\begin{array}{c}
\begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\
\begin{pmatrix}
e_{\mathcal{B}} & (-8, \leq) & (-4, \leq) \\
(10, \leq) & e_{\mathcal{B}} & (4, \leq) \\
(6, \leq) & (-4, \leq) & e_{\mathcal{B}}
\end{pmatrix}
\end{array}
\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}
$$

Therefore, image of X_0 w.r.t the component $\boldsymbol{g} = (1,1)$ is $X_1|_{\boldsymbol{g}=(1,1)} = \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 10,\ 4 \leq x_2' \leq 6,\ x_1' - x_2' = 4\}$. Applying the same procedure for $D^{(1,2)}$ and $D^{(2,2)}$ we obtain:

$$
D^{(X_1|_{\boldsymbol{g}=(1,2)})} =
\begin{array}{c}
\begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\
\begin{pmatrix}
e_{\mathcal{B}} & (-8, \leq) & (-4, \leq) \\
(10, \leq) & e_{\mathcal{B}} & (4, \leq) \\
(8, \leq) & (-2, \leq) & e_{\mathcal{B}}
\end{pmatrix}
\end{array}
\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}
$$

$$
D^{(X_1|_{\boldsymbol{g}=(2,2)})} =
\begin{array}{c}
\begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\
\begin{pmatrix}
e_{\mathcal{B}} & (-8, \leq) & (-6, \leq) \\
(11, \leq) & e_{\mathcal{B}} & (2, \leq) \\
(9, \leq) & (-2, \leq) & e_{\mathcal{B}}
\end{pmatrix}
\end{array}
\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}
$$

Thus, $X_1|_{\boldsymbol{g}=(1,2)} = \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 10,\ 4 \leq x_2' \leq 8,\ 2 \leq x_1' - x_2' \leq 4\}$ *and* $X_1|_{\boldsymbol{g}=(2,2)} = \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 11,\ 6 \leq x_2' \leq 9,\ x_1' - x_2' = 2\}$*. The reach set* X_1 *is the union of the images of* X_0 *w.r.t. each component of PWA system, i.e.,* $X_1 = X_1|_{\boldsymbol{g}=(1,1)} \cup X_1|_{\boldsymbol{g}=(1,2)} \cup X_1|_{\boldsymbol{g}=(2,2)} = \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 10,\ 4 \leq x_2' \leq 8,\ 2 \leq x_1' - x_2' \leq 4\} \cup \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 11,\ 6 \leq x_2' \leq 9,\ x_1' - x_2' = 2\}$*.*

Note that $D^{(X_1|_{\boldsymbol{g}=(1,1)})} = D^{(X_1|_{\boldsymbol{g}=(1,1)})} \oplus_{\mathcal{B}} D^{(X_1|_{\boldsymbol{g}=(1,2)})}$*, thus* $D^{(X_1|_{\boldsymbol{g}=(1,1)})} \cup D^{(X_1|_{\boldsymbol{g}=(1,2)})} = D^{(X_1|_{\boldsymbol{g}=(1,2)})}$ *(see remark 2.22). Therefore the reach set* X_1 *is represented by the collection of DBM given by* $\mathtt{D}^{(X_1)} = \{D^{(X_1|_{\boldsymbol{g}=(1,2)})},\ D^{(X_1|_{\boldsymbol{g}=(2,2)})}\}$*.*

The reach set X_2 *is obtained by computing the image of each DBM in* $\mathtt{D}^{(X_1)}$ *w.r.t each DBM in* $\mathtt{D}^{(PWA)} = \{D^{(1,1)},\ D^{(1,2)},\ D^{(2,2)}\}$*, which yields* $X_2 = \{\boldsymbol{x}' \in \mathbb{R}^2 : 16 \leq x_1' \leq 19,\ 12 \leq x_2' \leq 15,\ x_1' - x_2' = 4\}$*.*

Moreover, we observe that the system matrix has eigenvalue $\lambda = 8$ *and cyclicity* $c = 1$*, and for* $k \geq 2$*, we obtain* $X_{k+1} = 8 \otimes X_k$*. Thus, the reach set* X_3 *is simply obtained by computing* $8 \otimes X_2$*, which yields* $X_3 = \{\boldsymbol{x}' \in \mathbb{R}^2 : 24 \leq x_1' \leq 27,\ 20 \leq x_2' \leq 23,\ x_1' - x_2' = 4\}$*. The reach tube for* $k \in \{1, 2, 3\}$ *is shown in Figure 6.*

To compute the backward reach set $X_{-1} = \mathcal{I}_A^{-1}\{X_0\}$*, we must to compute the inverse image of* X_0 *w.r.t each component* $\boldsymbol{g}$ *of the PWA system. According to algorithm 3.2, the inverse image of* $D^{(X_0)}$ *w.r.t.* $D^{(1,1)}$ *can be computed as follows: first, we compute the cartesian product of* $D^{(X_0)}$ *and* $D^{(\mathbb{R}^2)}$*:*

$$
D^{(X_0 \times \mathbb{R}^2)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccccc} e_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (6, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}
$$

Then, we compute the intersection of $D^{(X_0 \times \mathbb{R}^2)}$ *and* $D^{(1,1)}$*:*

$$
D^{(X_0 \times \mathbb{R}^2)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccccc} e_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (8, \leq) & \varepsilon_{\mathcal{B}} \\ (6, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-8, \leq) & (-4, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (1, \leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}
$$

Next, we compute the canonical form representation of the intersection:

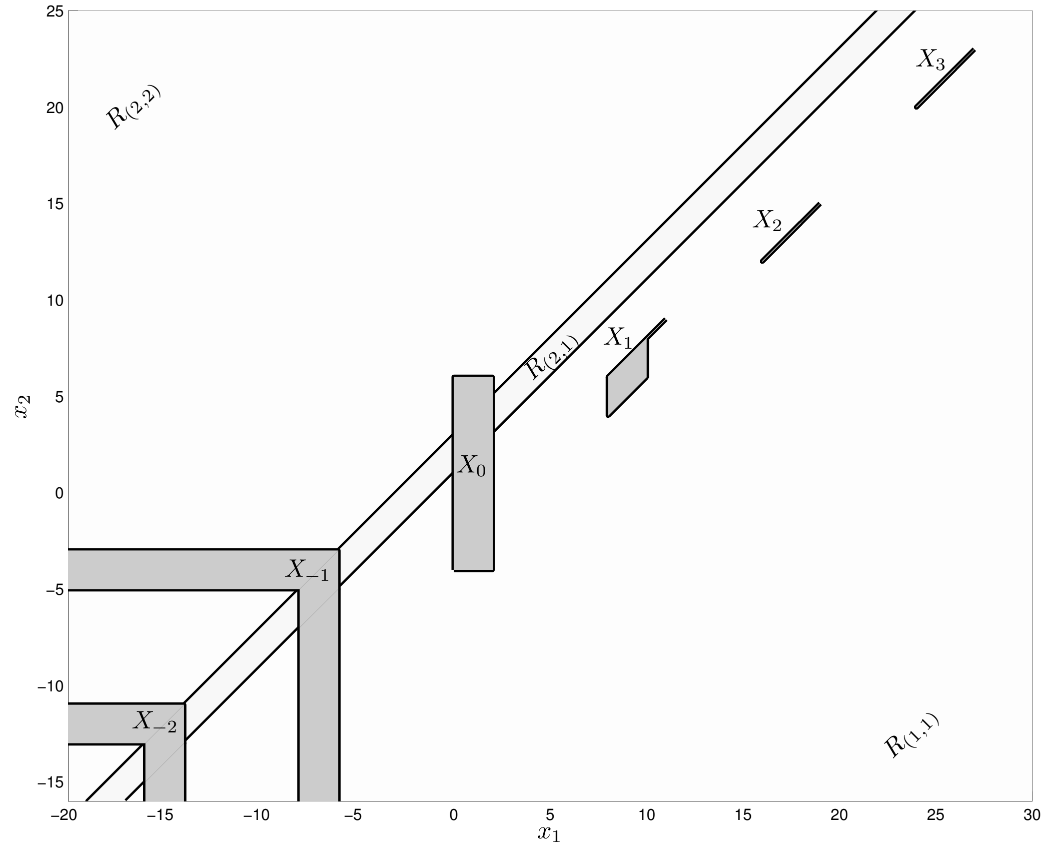

Figure 6 – reach tube for $k = \{1, 2, 3\}$ and backward reach tube for $k = \{1, 2\}$.

$$
cf(D^{(X_0 \times \mathbb{R}^2)} \oplus_{\mathcal{B}} D^{(1,1)}) = \begin{array}{cc} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (4, \leq) & (8, \leq) & \varepsilon_{\mathcal{B}} \\ (2, \leq) & e_{\mathcal{B}} & (4, \leq) & (8, \leq) & \varepsilon_{\mathcal{B}} \\ (-2, \leq) & (-4, \leq) & e_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\ (-6, \leq) & (-8, \leq) & (-4, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-5, \leq) & (-7, \leq) & (-3, \leq) & (1, \leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array} \end{array}
$$

Finally, we compute the orthogonal projection of the canonical form over the variables x_1 and x_2:

$$D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})} = cf(D^{(X_0\times\mathbb{R}^2)} \oplus_{\mathcal{B}} D^{(1,1)})\lceil_{\boldsymbol{x}} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (8,\leq) & \varepsilon_{\mathcal{B}} \\ (-6,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-5,\leq) & (1,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

Therefore, inverse image of X_0 *w.r.t the component* $\boldsymbol{g} = (1,1)$ *is* $X_{-1}|_{\boldsymbol{g}=(1,1)} = \{\boldsymbol{x} \in \mathbb{R}^2 : -8 \leq x_1 \leq -6,\ x_2 \leq -5,\ x_1 - x_2 \geq -1\}$. *Applying the same procedure for* $D^{(1,2)}$ *and* $D^{(2,2)}$ *we obtain:*

$$D^{(X_{-1}|_{\boldsymbol{g}=(1,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (8,\leq) & (7,\leq) \\ (-6,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ (-3,\leq) & (3,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

$$D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) \\ (-6,\leq) & e_{\mathcal{B}} & (-3,\leq) \\ (-3,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

Thus, $X_{-1}|_{\boldsymbol{g}=(1,2)} = \{\boldsymbol{x} \in \mathbb{R}^2 : -8 \leq x_1 \leq -6,\ -7 \leq x_2 \leq -3,\ -3 \leq x_1 - x_2 \leq -1\}$ *and* $X_{-1}|_{\boldsymbol{g}=(2,2)} = \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \leq -6,\ -5 \leq x_2 \leq -3,\ x_1 - x_2 \leq -3\}$. *The backward reach set* X_{-1} *is the union of the inverse images of* X_0 *w.r.t. each component of PWA system, i.e.,* $X_{-1} = X_{-1}|_{\boldsymbol{g}=(1,1)} \cup X_{-1}|_{\boldsymbol{g}=(1,2)} \cup X_{-1}|_{\boldsymbol{g}=(2,2)}$. *Observe that* X_{-1} *is represented by the collection of DBM given by* $\mathsf{D}^{(X_{-1})} = \{D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})},\ D^{(X_{-1}|_{\boldsymbol{g}=(1,2)})},\ D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})}\}$.

The backward reach set X_{-2} *is obtained by computing the inverse image of each DBM in* $\mathsf{D}^{(X_{-1})}$ *w.r.t each DBM in* $\mathsf{D}^{(PWA)} = \{D^{(1,1)}, D^{(1,2)}, D^{(2,2)}\}$, *which yields* $X_{-2} = \{\boldsymbol{x} \in \mathbb{R}^2 : -16 \leq x_1 \leq -14,\ x_2 \leq -13,\ x_1 - x_2 \geq -1\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : -16 \leq x_1 \leq -14,\ -15 \leq x_2 \leq -11,\ -3 \leq x_1 - x_2 \leq -1\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \leq -14,\ -13 \leq x_2 \leq -11,\ x_1 - x_2 \leq -3\}$.

4 Uncertain Max-Plus Linear Systems

As presented in section 2.3, the MPL systems matrices are associated to system delays and transport times. In practice, these parameters may be subjected to noise and disturbances, which should be taken into account in order to avoid tracking error or closed loop instability (van den Boom; De Schutter, 2002). In general, these perturbations are max-plus-multiplicative and appear as uncertainties in the max-plus model parameters. The Stochastic Max-Plus Linear (SMPL) systems are defined as MPL systems where the matrices entries are characterized by random variables (OLSDER *et al.*, 1990; RESING *et al.*, 1990; HEIDERGOTT, 2006; van den Boom; De Schutter, 2002; DILORETO *et al.*, 2010; HARDOUIN *et al.*, 2010). In this work, although the stochastic systems are not considered[1], we are interested in systems where the uncertain parameters can vary over a known interval. Formally, we define the uncertain Max-Plus Linear (uMPL) systems as nondeterministic MPL systems where, at each event step, the entries of the system matrices can, independently, take an arbitrary value within an real interval.

The autonomous model of an uMPL system is given by:

$$\mathbf{x}(k) = A(k) \otimes \mathbf{x}(k-1), \tag{4.1}$$

where the entries of $A(k) \in \overline{\mathbb{R}}_{max}^{n\times n}$ are considered to be in a real interval at each event step k, i.e., $a_{ij}(k) \in [\underline{a}_{ij},\ \overline{a}_{ij}]$.

Remark 4.1 *To assure the FIFO (first in, first out) rule the matrix $A(k)$ must satisfy $A(k) \succeq e$.*

Example 4.2 *In the public transport system of example 2.8, the travel times are assumed to be fixed. Now, let us consider that the travel times are in a real interval as indicated on the graph of figure 7.*

The system can be described by the following uMPL system:

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1),\ \textit{where}\ A(k) \in \begin{pmatrix} [2,\ 3] & [5,\ 6] \\ [3,\ 4] & [3,\ 4] \end{pmatrix}.$$

The nonautonomous model of an uMPL system is given by:

$$\mathbf{x}(k) = A(k) \otimes \mathbf{x}(k-1) \oplus B(k) \otimes \mathbf{u}(k), \tag{4.2}$$

[1] The probabilistic aspects of the uncertainties are not considered.

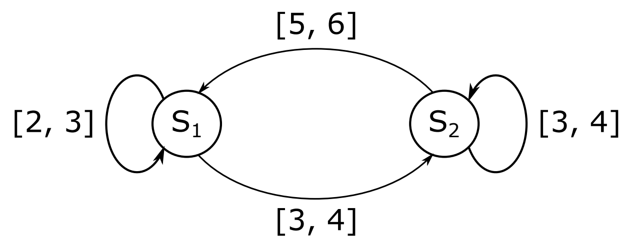

Figure 7 – Railway network model with uncertain travel times.

where the entries of where $A(k) \in \overline{\mathbb{R}}^{n\times n}_{max}$ and $B(k) \in \overline{\mathbb{R}}^{n\times m}_{max}$ are considered to be in a real interval at each event step k.

Equivalently to the deterministic case, any nonautonomous uMPL system can be transformed into an augmented autonomous uMPL model by considering $F(k) = (A(k)\ B(k)) \in \bar{R}^{n\times(n+m)}_{max}$ and $\mathbf{y}(k-1) = \left(\mathbf{x}(k-1)^T\ \mathbf{u}(k)^T\right)^T$.

$$\mathbf{x}(k) = F(k) \otimes \mathbf{y}(k-1). \tag{4.3}$$

4.1 Interval Analysis

This section introduces some basic concepts of Interval Analysis and its applications to the uMPL systems (MOORE; BIERBAUM, 1979; LITVINOV; SOBOLEVSKIĪ, 2001; GNING *et al.*, 2012; BRUNSCH *et al.*, 2012; HARDOUIN *et al.*, 2009; LHOMMEAU *et al.*, 2005).

An interval is defined as a closed set of real numbers (MOORE; BIERBAUM, 1979):

$$[\mathbf{x}] = [\underline{x},\ \overline{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}\}. \tag{4.4}$$

A degenerate interval is an interval consisting of a single real number. Thus, an interval $[\mathbf{x}]$ is degenerate if $\underline{x} = \overline{x}$.

Remark 4.3 *By convention a degenerate interval $[x,\ x]$ is identified with the real number x.*

Remark 4.4 *The notation $[\boldsymbol{X}]$ will be used for matrices of intervals, i.e., matrices whose entries are intervals:*

$$[\boldsymbol{X}] = [\underline{X},\ \overline{X}] = ([\boldsymbol{x}_{ij}])_{\substack{1\leq i\leq n \\ 1\leq j\leq p}}. \tag{4.5}$$

The *intersection* and *union* operations can be applied to intervals (MOORE; BIERBAUM, 1979). The intersection of two intervals $[\mathbf{x}]$ and $[\mathbf{y}]$ is always an interval, defined by:

$$[\mathbf{x}] \cap [\mathbf{y}] = [\max\{\underline{x}, \underline{y}\},\ \min\{\overline{x}, \overline{y}\}]. \tag{4.6}$$

Thus, the intersection is empty if either $\underline{x} > \overline{y}$ or $\overline{x} < \underline{y}$. On the other hand, the *union* of two intervals is *not*, in general, an interval. However, if the intervals have nonempty intersection, their union is again an interval defined by:

$$[\mathbf{x}] \cup [\mathbf{y}] = [\min\{\underline{x}, \underline{y}\},\ \max\{\overline{x}, \overline{y}\}]. \tag{4.7}$$

Example 4.5 *Consider the intervals:* $[\boldsymbol{x}] = [0,\ 4]$, $[\boldsymbol{y}] = [2,\ 5]$ *and* $[\boldsymbol{z}] = [5,\ 7]$. *Then,*

$$\begin{aligned} [\boldsymbol{x}] \cap [\boldsymbol{y}] &= [\max\{0,2\},\ \min\{4,5\}] = [2,\ 4], \\ [\boldsymbol{x}] \cap [\boldsymbol{z}] &= [\max\{0,5\},\ \min\{4,7\}] = [5,\ 4] = \emptyset, \\ [\boldsymbol{y}] \cap [\boldsymbol{z}] &= [\max\{2,5\},\ \min\{5,7\}] = [5,\ 5]. \end{aligned}$$

Since $[\boldsymbol{x}] \cap [\boldsymbol{y}]$ *and* $[\boldsymbol{y}] \cap [\boldsymbol{z}]$ *are not empty we have that:*

$$\begin{aligned} [\boldsymbol{x}] \cup [\boldsymbol{y}] &= [\min\{0,2\},\ \max\{4,5\}] = [0,\ 5], \\ [\boldsymbol{x}] \cup [\boldsymbol{z}] &= [0,\ 4] \cup [5,\ 7], \\ [\boldsymbol{y}] \cup [\boldsymbol{z}] &= [\min\{2,5\},\ \max\{5,7\}] = [2,\ 7]. \end{aligned}$$

Note that the intersection of $[\boldsymbol{x}]$ *and* $[\boldsymbol{z}]$ *is empty, and therefore* $[\boldsymbol{x}] \cup [\boldsymbol{z}]$ *is not an interval.*

The binary operations + and - can be extended to intervals (MOORE; BIERBAUM, 1979):

$$[\mathbf{x}] + [\mathbf{y}] = \{x + y : x \in [\mathbf{x}]\,,\ y \in [\mathbf{y}]\} = [\underline{x} + \underline{y},\ \overline{x} + \overline{y}], \tag{4.8}$$

$$[\mathbf{x}] - [\mathbf{y}] = \{x - y : x \in [\mathbf{x}]\,,\ y \in [\mathbf{y}]\} = [\underline{x} - \overline{y},\ \overline{x} - \underline{y}]. \tag{4.9}$$

Example 4.6 *Let* $[\boldsymbol{x}] = [4,\ 8]$ *and* $[\boldsymbol{y}] = [3,\ 5]$. *Then*

$$\begin{aligned} [\boldsymbol{x}] + [\boldsymbol{y}] &= [4+3,\ 8+5] = [7,\ 13], \\ [\boldsymbol{x}] - [\boldsymbol{y}] &= [4-5,\ 8-3] = [-1,\ 5]. \end{aligned}$$

We can also extend the max-plus operations to intervals (BRUNSCH *et al.*, 2012; HARDOUIN *et al.*, 2009; LHOMMEAU *et al.*, 2005):

$$[\mathbf{x}] \oplus [\mathbf{y}] = \ \{x \oplus y : x \in [\mathbf{x}]\,,\ y \in [\mathbf{y}]\} = [\underline{x} \oplus \underline{y},\ \overline{x} \oplus \overline{y}], \tag{4.10}$$

$$[\mathbf{x}] \otimes [\mathbf{y}] = \ \{x \otimes y : x \in [\mathbf{x}]\,,\ y \in [\mathbf{y}]\} = [\underline{x} \otimes \underline{y},\ \overline{x} \otimes \overline{y}]. \tag{4.11}$$

Moreover, if $[\mathbf{A}]$, $[\mathbf{B}]$ and $[\mathbf{C}]$ are $n\times p$, $n\times p$ and $p\times q$ matrices of intervals, respectively, we have that:

$$\begin{aligned}
([\mathbf{A}] \oplus [\mathbf{B}])_{ij} &= [\mathbf{a}_{ij}] \oplus [\mathbf{b}_{ij}] \\
&= [\underline{a_{ij}} \oplus \underline{b_{ij}},\ \overline{a_{ij}} \oplus \overline{b_{ij}}], \qquad (4.12)\\
([\mathbf{A}] \otimes [\mathbf{C}])_{ij} &= \bigoplus_{k=1}^{p} ([\mathbf{a}_{ik}] \otimes [\mathbf{c}_{kj}]) \\
&= \bigoplus_{k=1}^{p} \left\{ [\underline{a}_{ik} \otimes \underline{c}_{kj},\ \overline{a}_{ik} \otimes \overline{c}_{kj}] \right\} \\
&= \left[\bigoplus_{k=1}^{p} \left\{ \underline{a}_{ik} \otimes \underline{c}_{kj} \right\},\ \bigoplus_{k=1}^{p} \{ \overline{a}_{ik} \otimes \overline{c}_{kj} \} \right], \qquad (4.13)
\end{aligned}$$

or, equivalently:

$$[\mathbf{A}] \oplus [\mathbf{B}] = [\underline{A} \oplus \underline{B},\ \overline{A} \oplus \overline{B}], \qquad (4.14)$$

$$[\mathbf{A}] \otimes [\mathbf{C}] = [\underline{A} \otimes \underline{C},\ \overline{A} \otimes \overline{C}]. \qquad (4.15)$$

Thus, the k^{th} power of a matrix of intervals is given by:

$$[\mathbf{A}]^{\otimes k} = [\underline{A}^{\otimes k},\ \overline{A}^{\otimes k}]. \qquad (4.16)$$

A partial order for intervals in $\overline{\mathbb{R}}_{max}$ can be defined as:

$$[\mathbf{x}] \succeq [\mathbf{y}] \Leftrightarrow [\mathbf{x}] = [\mathbf{x}] \oplus [\mathbf{y}] \Leftrightarrow \underline{x} \succeq \underline{y} \text{ and } \overline{x} \succeq \overline{y}. \qquad (4.17)$$

In particular,

$$[\mathbf{x}] = [\mathbf{y}] \Leftrightarrow \underline{x} = \underline{y} \text{ and } \overline{x} = \overline{y}. \qquad (4.18)$$

Moreover, the max-plus sum can be extended to a finitely many number of intervals:

$$\bigoplus_{i=1}^{n} [\mathbf{x}]_i = \left\{ \bigoplus_{i=1}^{n} x_i : x_i \in [\mathbf{x}_i] \right\} = \left[\bigoplus_{i=1}^{n} \underline{x}_i,\ \bigoplus_{i=1}^{n} \overline{x}_i \right]. \qquad (4.19)$$

Consider now a generic uMPL system given by:

$$\mathbf{z}(k) = A(k) \otimes \mathbf{x}(k-1),\ A(k) \in [\mathbf{A}] \qquad (4.20)$$

where $\mathbf{z}(k) \in \overline{\mathbb{R}}^n_{max}$ and $\mathbf{x}(k-1) \in \overline{\mathbb{R}}^p_{max}$.

The i-th equation of (4.20) can be expressed as:

$$z_i(k) = \bigoplus_{j=1}^{p} \{a_{ij}(k) \otimes x_j(k-1)\}, \quad a_{ij}(k) \in [\mathbf{a}_{ij}]\,. \qquad (4.21)$$

Therefore, given $\mathbf{x}(k-1)$, and by using equations (4.11) and (4.19), $z_i(k)$ is in the interval defined by:

$$\begin{aligned}[\mathbf{z}_i](k) &= \bigoplus_{j=1}^{p}\{[\mathbf{a}_{ij}](k) \otimes x_j(k-1)\} \\ &= \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\},\ \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j(k-1)\}\right].\end{aligned} \tag{4.22}$$

Example 4.7 *Consider the following uMPL system:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1), \quad A(k) \in [\boldsymbol{A}],$$

where

$$[\boldsymbol{A}] = \begin{pmatrix} [2,\ 7] & [4,\ 5] \\ [4,\ 6] & [2,\ 6] \end{pmatrix}.$$

Given $\boldsymbol{x}(0) = (0\ 1)^T$*, then* $\boldsymbol{x}(1) \in [\boldsymbol{x}](1) = ([\boldsymbol{x}_1](1)\ \ [\boldsymbol{x}_2](1))^T$*, where:*

$$\begin{pmatrix} [\boldsymbol{x}_1](1) \\ [\boldsymbol{x}_1](2) \end{pmatrix} = \begin{pmatrix} [(2 \otimes 0) \oplus (4 \otimes 1),\ (7 \otimes 0) \oplus (5 \otimes 1)] \\ [(4 \otimes 0) \oplus (2 \otimes 1),\ (6 \otimes 0) \oplus (6 \otimes 1)] \end{pmatrix} = \begin{pmatrix} [5,\ 7] \\ [4,\ 7] \end{pmatrix}.$$

4.2 Partitioned Uncertain MPL systems

This section presents the main contribution of this work. We aim to use the DBM data structure for the reachability analysis of uMPL systems. In Section 2.5.1 we have seen that every MPL system can be expressed as a PWA system and Chapter 3 shows how DBM representation of PWA systems is efficient for reachability analysis. Seeking for generality, we observe that the reachability analysis of an MPL system through the DBM approach is possible because each affine system (2.55) and its corresponding active state space region (2.54) can be independently represented by one DBM. In the following, we propose a partition of the state space of uMPL systems that satisfies this property. On this purpose let us express interval (4.22) as:

$$z_i \in \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\},\ \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j\}\right] \Longleftrightarrow \bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\} \preceq z_i \preceq \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j\} \tag{4.23}$$

Observe that $\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\} \preceq z_i$ can be alternatively expressed as $\bigcap_{j=1}^{p}\{x_j(k-1) - z_i(k) \leq -\underline{a}_{ij}\}$, and therefore the lower bound of (4.23) can be depicted in a single DBM. On the other hand, the term $z_i \preceq \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j\}$ is equivalent to $\bigcup_{j=1}^{p}\{z_i(k) - x_j(k-1) \leq \overline{a}_{ij}\}$.

Note that, each term of this union can be represented by a DBM. However, in general, the union of DBM is not a DBM. Therefore, the upper bound of (4.23) cannot be depicted in a single DBM. The main contribution of this work is to propose a partition of the state space in which (4.23) can be expressed as a DBM suitable form.

Following the later arguments, we must search for regions in which the upper bound of (4.23) can be expressed as a DBM. Then, let us consider the problem of finding the region where $[\mathbf{z}_i]$ can be expressed as:

$$[\mathbf{z}_i] = \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\}, \quad \overline{a}_{ig_i} \otimes x_{g_i}\right] \quad \forall i \in \{1, ..., n\}, \tag{4.24}$$

where $\mathbf{g} = (g_1, \cdots, g_n) \in \{1, ..., p\}^n$ has the same interpretation as in (2.54).

This problem corresponds to find a region where the following equality holds:

$$\left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\},\ \overline{a}_{ig_i} \otimes x_{g_i}\right] = \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\},\ \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j\}\right] \quad \forall i \in \{1, ..., n\}. \tag{4.25}$$

From (4.18), the equality holds if:

$$\overline{a}_{ig_i} \otimes x_{g_i} = \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j\} \quad \forall i \in \{1, ..., n\}. \tag{4.26}$$

According to (2.33), equation (4.26) can be expressed as:

$$\overline{a}_{ig_i} \otimes x_{g_i} \succeq \overline{a}_{ij} \otimes x_j \quad \forall i, j, \tag{4.27}$$

which is equivalent to:

$$x_j - x_{g_i} \leq \overline{a}_{ig_i} - \overline{a}_{ij} \quad \forall i, j. \tag{4.28}$$

The region corresponding to (4.28) is given by:

$$R^u_{\mathbf{g}} = \bigcap_{i=1}^{n} \bigcap_{\substack{j=1 \\ j \neq g_i}}^{p} \left\{\mathbf{x} \in \overline{\mathbb{R}}^p_{max} : x_j - x_{g_i} \leq \overline{a}_{ig_i} - \overline{a}_{ij}\right\}. \tag{4.29}$$

Region (4.29) defines a partition for uMPL systems. Moreover, if $\mathbf{x} \in R^u_{\mathbf{g}}$ then $z_i(k)$ is in the interval defined in (4.24). i.e.,

$$z_i \in \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j\}, \quad \overline{a}_{ig_i} \otimes x_{g_i}\right] \ \forall i, \text{ if } \mathbf{x} \in R^u_{\mathbf{g}}. \tag{4.30}$$

Example 4.8 *Consider the following autonomous uMPL system:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1), \ \ \textit{where } A(k) \in \begin{pmatrix} [4,\ 6] & [3,\ 5] \\ [3,\ 7] & [4,\ 5] \end{pmatrix}.$$

According to equation (4.29)*, the regions corresponding to each component* $\boldsymbol{g} \in \{1,2\}^2 = \{(1,1),\ (1,2),\ (2,1),\ (2,2)\}$ *are given by:*

$$\begin{aligned}
R^u_{(1,1)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 1\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 2\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 1\right\}, \\
R^u_{(1,2)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 1\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -2\right\} \\
&= \emptyset, \\
R^u_{(2,1)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -1\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \leq 2\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 1 \leq x_2 - x_1 \leq 2\right\}, \\
R^u_{(2,2)} &= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -1\right\} \cap \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_1 - x_2 \leq -2\right\} \\
&= \left\{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : x_2 - x_1 \geq 2\right\}.
\end{aligned}$$

Then, according to (4.30) *the corresponding partitioned uMPL system is*[2]*:*

$$\boldsymbol{x}' \in \begin{cases}
\begin{pmatrix} [(4 \otimes x_1) \oplus (3 \otimes x_2)\,,\ 6 \otimes x_1] \\ [(3 \otimes x_1) \oplus (4 \otimes x_2)\,,\ 7 \otimes x_1] \end{pmatrix} & \textit{if} \quad \boldsymbol{x} \in R^u_{(1,1)}, \\
\begin{pmatrix} [(4 \otimes x_1) \oplus (3 \otimes x_2)\,,\ 5 \otimes x_2] \\ [(3 \otimes x_1) \oplus (4 \otimes x_2)\,,\ 7 \otimes x_1] \end{pmatrix} & \textit{if} \quad \boldsymbol{x} \in R^u_{(2,1)}, \\
\begin{pmatrix} [(4 \otimes x_1) \oplus (3 \otimes x_2)\,,\ 5 \otimes x_2] \\ [(3 \otimes x_1) \oplus (4 \otimes x_2)\,,\ 5 \otimes x_2] \end{pmatrix} & \textit{if} \quad \boldsymbol{x} \in R^u_{(2,2)},
\end{cases}$$

Figure 8 depicts the generated partitioned uMPL.

Example 4.9 *Consider the following nonautonomous uMPL system:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1) \oplus B(k) \otimes \boldsymbol{u}(k),$$

where,

$$A(k) \in \begin{pmatrix} 2 & [2,\ 4] \\ [3,\ 5] & [3,\ 4] \end{pmatrix} \ \ \textit{and } B(k) \in \begin{pmatrix} e \\ \varepsilon \end{pmatrix}.$$

[2] Notation: $\mathbf{x}' \equiv \mathbf{x}(k)$ and $\mathbf{x} \equiv \mathbf{x}(k-1)$.

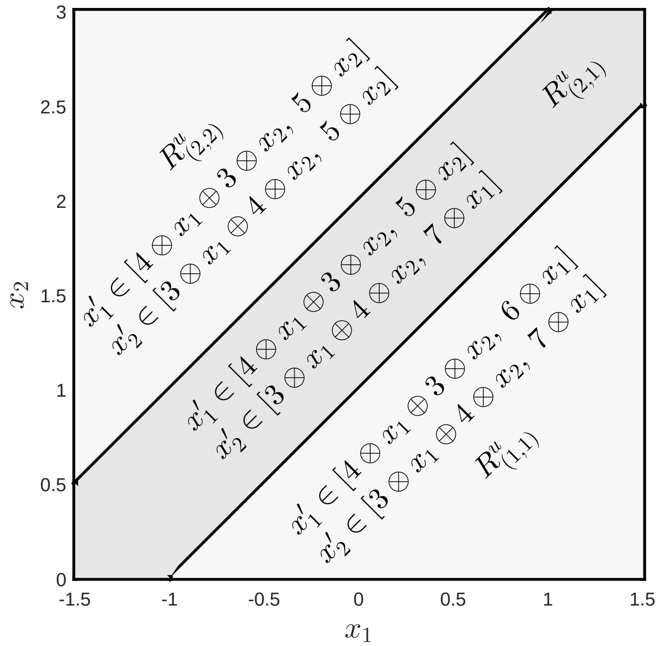

Figure 8 – A partitioned uMPL system.

This system can be expressed as the following augmented autonomous uMPL system:

$$\boldsymbol{x}(k) = F(k) \otimes \boldsymbol{y}(k-1), \text{ where } \boldsymbol{y}(k-1) = \begin{pmatrix} x_1(k-1) \\ x_2(k-1) \\ u_1(k) \end{pmatrix} \text{ and } F(k) \in \begin{pmatrix} 2 & [2,\ 4] & e \\ [3,\ 5] & [3,\ 4] & \varepsilon \end{pmatrix}.$$

In order to express the uMPL system as a partitioned uMPL system we must compute the regions corresponding to each component $\boldsymbol{g} \in \{1,2,3\}^2 = \{(1,1),\ (1,2),\ (1,3),\ (2,1),\ (2,2),\ (2,3),\ (3,1),\ (3,2),\ (3,3)\}$*. Since the matrix entry* $[\boldsymbol{f}_{23}]$ *is null (in the max-plus sense) we have that the regions corresponding to the components* $\{(1,3),\ (2,3),\ (3,3)\}$ *are empty. According to equation* (4.29)*, the regions corresponding to the components* $\boldsymbol{g} \in \{(1,1),\ (1,2),\ (2,1),\ (2,2),\ (3,1),\ (3,2)\}$ *are given by:*

$$
\begin{aligned}
R^u_{(1,1)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq 2\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq 1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq 2\right\}, \\
R^u_{(1,2)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq 2\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq \infty\right\} \\
&= \emptyset, \\
R^u_{(2,1)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq 2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq 4\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq 1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : -2 \leq y_2 - y_1 \leq 1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq 4\right\}, \\
R^u_{(2,2)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq 2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq 4\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq 4\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -1\right\}, \\
R^u_{(3,1)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \\
&\qquad \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq 1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_1 \leq \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_1 \leq 1\right\}, \\
R^u_{(3,2)} &= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \\
&\qquad \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -1\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_3 - y_2 \leq \infty\right\} \\
&= \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_3 \leq -2\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_2 - y_3 \leq -4\right\} \cap \left\{\boldsymbol{y} \in \overline{\mathbb{R}}^3_{max} : y_1 - y_2 \leq -1\right\}.
\end{aligned}
$$

Then, according to (4.30) *the corresponding partitioned uMPL system is*[3]

$$
\boldsymbol{x}' \in \begin{cases}
\begin{pmatrix} [(2 \otimes y_1) \oplus (2 \otimes y_2) \oplus (e \otimes y_3)\,,\ 2 \otimes y_1] \\ [(3 \otimes y_1) \oplus (3 \otimes y_2) \oplus (\varepsilon \otimes y_3)\,,\ 5 \otimes y_1] \end{pmatrix} & \textit{if} \quad \boldsymbol{y} \in R^u_{(1,1)}, \\
\begin{pmatrix} [(2 \otimes y_1) \oplus (2 \otimes y_2) \oplus (e \otimes y_3)\,,\ 4 \otimes y_2] \\ [(3 \otimes y_1) \oplus (3 \otimes y_2) \oplus (\varepsilon \otimes y_3)\,,\ 5 \otimes y_1] \end{pmatrix} & \textit{if} \quad \boldsymbol{y} \in R^u_{(2,1)}, \\
\begin{pmatrix} [(2 \otimes y_1) \oplus (2 \otimes y_2) \oplus (e \otimes y_3)\,,\ 4 \otimes y_2] \\ [(3 \otimes y_1) \oplus (3 \otimes y_2) \oplus (\varepsilon \otimes y_3)\,,\ 4 \otimes y_2] \end{pmatrix} & \textit{if} \quad \boldsymbol{y} \in R^u_{(2,2)}, \\
\begin{pmatrix} [(2 \otimes y_1) \oplus (2 \otimes y_2) \oplus (e \otimes y_3)\,,\ e \otimes y_3] \\ [(3 \otimes y_1) \oplus (3 \otimes y_2) \oplus (\varepsilon \otimes y_3)\,,\ 5 \otimes y_1] \end{pmatrix} & \textit{if} \quad \boldsymbol{y} \in R^u_{(3,1)}, \\
\begin{pmatrix} [(2 \otimes y_1) \oplus (2 \otimes y_2) \oplus (e \otimes y_3)\,,\ e \otimes y_3] \\ [(3 \otimes y_1) \oplus (3 \otimes y_2) \oplus (\varepsilon \otimes y_3)\,,\ 4 \otimes y_2] \end{pmatrix} & \textit{if} \quad \boldsymbol{y} \in R^u_{(3,2)},
\end{cases}
$$

[3] Notation: $\mathbf{x}' \equiv \mathbf{x}(k)$ and $\mathbf{y} \equiv \mathbf{y}(k-1)$.

4.2.1 DBM Representation of Partitioned uMPL systems

Each region (4.29) can be represented by a $(p+1)\times(p+1)$ DBM, see Section 2.4. From (4.24), $z_i(k)$, $i \in \{1,...,n\}$, is in the set defined by the following inequalities:

$$z_i(k) \preceq \overline{a}_{ig_i} \otimes x_{g_i}(k-1), \tag{4.31}$$

$$z_i(k) \succeq \bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\} \Leftrightarrow \begin{cases} z_i(k) \succeq \underline{a}_{i1} \otimes x_1(k-1), \\ \vdots \\ z_i(k) \succeq \underline{a}_{ip} \otimes x_p(k-1). \end{cases} \tag{4.32}$$

From this set, the following region can be defined:

$$\bigcap_{i=1}^{n}\{z_i(k) - x_{g_i}(k-1) \leq \overline{a}_{ig_i}\} \cap \bigcap_{i=1}^{n}\bigcap_{\substack{j=1\\ j\neq i}}^{p}\left\{x_j(k-1) - z_i(k) \leq -\underline{a}_{ij}\right\} \tag{4.33}$$

Therefore, it is straightforward to see that the dynamics of a partitioned uMPL system can be represented by a $(n+p+1)\times(n+p+1)$ DBM.

Remark 4.10 *Each component of a partitioned uMPL system (region plus corresponding dynamics) can be fully characterized by the intersection of* (4.29) *and* (4.33)*. This intersection can be represented by a* $(n+p+1)\times(n+p+1)$ *DBM which constrains the variables* $[z_1,\dots,z_n,x_1,\dots,x_p]$ *and their differences.*

Given $[\mathbf{A}] = [\underline{A},\ \overline{A}]$, where $\underline{A}$, and $\overline{A} \in \overline{\mathbb{R}}^{n\times p}_{max}$, Algorithm 4.1 describes a procedure to generate a partitioned uMPL system represented by a collection of DBM `D`.

Algorithm 4.1 works as follows: In step 1 the output and auxiliary variables are initialized. Step 5 generates a $n\times p$ matrix $(dynInf)$[4] representing the bounds for the differences defined in the right side of intersection (4.33). As can be observed in (4.33), these differences does not depends on $\mathbf{g}$, therefore they can be calculated before the main loop (step 9). Then, for each $\mathbf{g}$: step 13 generates an $p\times n$ matrix $(dynSup)$[5] representing the bounds for the differences defined in the left side of intersection (4.33); step 16 computes the DBM representation of region $R^u_{\mathbf{g}}$; if the obtained DBM is not empty (step 21) the matrices $dynInf$, $dynSup$ and the region $R^u_{\mathbf{g}}$ are used to generate a DBM $D^{\mathbf{g}} \in \mathbb{R}^{n+p}$ (steps 22 to 25) and step 26 saves $D^{\mathbf{g}}$ in `D`.

4 Note that $dynInf$ is not a DBM because it is not a square matrix.
5 $dynSup$ is not a DBM.

Algorithm 4.1: Expressing an MPL system as a PWA system using DBM as data structure. The assignment $dbmEye(\cdot)$ generates a square matrix of specified dimension, with entries $d_{ij} = e_{\mathcal{B}}$ if $i = j$ and $d_{ij} = \varepsilon_{\mathcal{B}}$ if $i \neq j$. The assignment $dbmNull(\cdot, \cdot)$ generates a matrix of specified dimension, with entries $d_{ij} = \varepsilon_{\mathcal{B}}$.

---

input : $[\mathbf{A}] = [\underline{A},\ \overline{A}]$, where $\underline{A},\ \overline{A} \in \mathbb{R}_{max}^{n\times p}$
output: `D`

1 `D` $\leftarrow \emptyset$, $dynInf \leftarrow dbmNull(n, n)$;
2 **for all** $i \in \{1, ..., n\}$ **do**
3   **for all** $j \in \{1, ..., p\}$ **do**
4   **if** $\underline{A}[i, j] \neq \epsilon$ **then**
5             $dynInf[j, i] \leftarrow (-\underline{a}_{ij}, \leq)$`// represents` $z_i \geq x_j + \underline{a}_{ij}$
6   **end if**
7   **end for**
8 **end for**
9 **for all** $\boldsymbol{g} \in \{1, ..., p\}^n$ **do**
10     $R^u_{\mathbf{g}} \leftarrow dbmEye(n)$, $dynSup \leftarrow dbmNull(n, p)$;
11   **for all** $i \in \{1, ..., n\}$ **do**
12   **if** $a_{ig_i} \neq \epsilon$ **then**
13             $dynSup[i, g_i] \leftarrow (\overline{a}_{ig_i}, \leq)$ `// represents` $z_i \leq x_{g_i} + \overline{a}_{ig_i}$
14   **for all** $j \in \{1, ..., p\}$ **do**
15   **if** $a_{i,j} \neq \epsilon$ **then**
16                     $R^u_{\mathbf{g}}[i, g_i] \leftarrow (\min\left\{R^u_{\mathbf{g}}[i, g_i], \overline{a}_{ig_i} - \overline{a}_{ij}\right\}, \leq)$ `// define` $R^u_{\mathbf{g}}$`, see` (4.29)
17   **end if**
18   **end for**
19   **end if**
20   **end for**
21   **if** R^u_{g} *is not empty* **then**
22         $D^{\mathbf{g}} \leftarrow dbmEye(n + p + 1)$ `//`
23         $D^{\mathbf{g}}[2 : n + 1,\ n + 2 : n + p + 1] \leftarrow dynSup$ `//`
24         $D^{\mathbf{g}}[n + 2 : n + p + 1,\ 2 : n + 1] \leftarrow dynInf$ `//`
25         $D^{\mathbf{g}}[n + 2 : n + p + 1,\ n + 2 : n + p + 1] \leftarrow R^u_{\mathbf{g}}$ `//`
26   `D` $\leftarrow$ `D` $\cup \{D^{\mathbf{g}}\}$;
27   **end if**
28 **end for**

$$D^{\mathbf{g}} = \begin{array}{c} \begin{array}{ccc} x_0 & z_1 \ldots z_n & x_1 \ldots x_p \end{array} \\ \left(\begin{array}{c|c|c} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \ldots \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \ldots \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} \\ \vdots & e_{\mathcal{B}}^{n\times n} & \begin{array}{c} dunSup \\ \text{(step 23)} \end{array} \\ \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} \\ \vdots & \begin{array}{c} dunInf \\ \text{(step 24)} \end{array} & \begin{array}{c} R_{\mathbf{g}} \\ \text{(step 25)} \end{array} \\ \varepsilon_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ z_1 \\ \vdots \\ z_n \\ x_1 \\ \vdots \\ x_p \end{array}$$

The worst-case complexity is calculated as follows. The maximum number of iterations in steps 9, 11 and 14 is p^n, n and p respectively. The complexity of the checking for emptiness of a DBM is cubic w.r.t. its dimension (see section 2.4.1), thus the complexity of step 21 is constant and equal to $\mathcal{O}\left(p^3\right)$. Moreover, the number of iterations in steps 2 and 3 is constant and amounts to np. Thus, the worst-case complexity is $\mathcal{O}\left(p^n(np + n^3)\right)$. As for the classical case (section 2.5), the bottleneck resides in the worst-case cardinality of the collection of coefficients $\mathbf{g}$, given by p^n. It should be noted that the performance of the algorithm can also be improved by using the backtracking technique discussed at the end of section 2.5.

In order to test the efficiency of the approach an experiment was carried out: for each $n \in \{10, 12, 14, 16, 18, 20\}$ it was generated an $n \times n$ matrix $[\mathbf{A}]$ with exactly 2 non-ε entries randomly placed in each row. The upper bound of the non-ε entries was randomly generated between 1 and 100 and the lower bound was set to[6] 0. In table 2 are average number of regions and the average time to generate the DBM representation over 10 experiments. The experiments were run in a Intel Core i7-6700HQ CPU @ 2.60 GHz with 16 GB of memory.

Table 2 – computation time to partition an uMPL system (average over 10 experiments)

n	number of regions	time to generate the DBM representation
10	7.16×10^2	$0.17(s)$
12	2.92×10^3	$0.75\ (s)$
14	1.05×10^4	$3.05\ (s)$
16	4.66×10^4	$14.64\ (s)$
18	2.05×10^5	$71.82\ (s)$
20	6.13×10^5	$4.41\ (min)$

Example 4.11 *In this example, the uMPL system of example 4.8 is alternatively represented as a collection of DBM. For each* $\boldsymbol{g} \in \{1,2\}^2 = \{(1,1),\ (1,2),\ (2,1), (2,2)\}$*, we compute a DBM* $D^{\boldsymbol{g}}$ *which represents the region* $R^u_{\boldsymbol{g}}$ *and the corresponding dynamics. The DBM* $D^{(1,1)}$ *is constructed as follows. From* (4.29)*, we have that:* $R^u_{(1,1)} = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : \underbrace{x_2 - x_1 \leq 1}_{d^{(1,1)}_{54}}\}$.

And, from (4.33)*, the dynamics active in* $R^u_{(1,1)}$ *is given by:* $\underbrace{\{x'_1 - x_1 \leq 6\}}_{d^{(1,1)}_{24}} \cap \underbrace{\{x'_2 - x_1 \leq 7\}}_{d^{(1,1)}_{34}}$

$\cap \underbrace{\{x_1 - x'_1 \leq -4\}}_{d^{(1,1)}_{42}} \cap \underbrace{\{x_2 - x'_1 \leq -3\}}_{d^{(1,1)}_{52}} \cap \underbrace{\{x_1 - x'_2 \leq -3\}}_{d^{(1,1)}_{43}} \cap \underbrace{\{x_2 - x'_2 \leq -4\}}_{d^{(1,1)}_{53}}.$

Thus, $D^{(1,1)}$ *is given by:*

[6] Note that the complexity of the algorithm critically depends on the number of regions and the regions only depends on the upper bounds of the matrix entries then setting the lower bounds to 0 does not interfere in the results of the experiment.

$$D^{(1,1)} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \\ \end{array}$$

$$D^{(1,1)} = \begin{array}{cc} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (6,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (7,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array} \end{array}$$

Similarly, the DBM $D^{(2,1)}$ is constructed as follows. From (4.29), we have that: $R^u_{(2,1)} = \underbrace{\{x_1 - x_2 \leq -1\}}_{d^{(2,1)}_{45}} \cap \underbrace{\{x_2 - x_1 \leq 2\}}_{d^{(2,1)}_{54}}$. *And, from (4.33), the dynamics active in $R^u_{(2,1)}$ is given by:*

$$\underbrace{\{x'_1 - x_2 \leq 5\}}_{d^{(2,1)}_{25}} \cap \underbrace{\{x'_2 - x_1 \leq 7\}}_{d^{(2,1)}_{34}} \cap \underbrace{\{x_1 - x'_1 \leq -4\}}_{d^{(2,1)}_{42}} \cap \underbrace{\{x_2 - x'_1 \leq -3\}}_{d^{(2,1)}_{52}} \cap \underbrace{\{x_1 - x'_2 \leq -3\}}_{d^{(2,1)}_{43}} \cap \underbrace{\{x_2 - x'_1 \leq -4\}}_{d^{(2,1)}_{53}}.$$

Thus, $D^{(2,1)}$ is given by:

$$D^{(2,1)} = \begin{array}{cc} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (7,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & (-3,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & (2,\leq) & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array} \end{array}$$

Using the same procedure, we obtain:

$$D^{(2,2)} = \begin{array}{cc} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & (-3,\leq) & e_{\mathcal{B}} & (-2,\leq) \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array} \end{array}$$

Example 4.12 *The uMPL system of example 4.9 can be represented by the following collection of DBM.*

$$D^{(1,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & (-2, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

$$D^{(2,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & e_{\mathcal{B}} & (2, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4, \leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

$$D^{(2,2)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & e_{\mathcal{B}} & (-1, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4, \leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

$$D^{(3,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-2, \leq) \\ \varepsilon_{\mathcal{B}} & (-2, \leq) & (-3, \leq) & (1, \leq) & e_{\mathcal{B}} & (-4, \leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

$$D^{(3,2)} = \begin{array}{cc} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} & \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & (-1,\leq) & (-2,\leq) \\ \varepsilon_{\mathcal{B}} & (-2,\leq) & (-3,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-4,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} & \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array} \end{array}$$

Remark 4.13 *If $\underline{a}_{ij} = \overline{a}_{ij}$ $\forall i,j$ (deterministic case), region $R^u_{\boldsymbol{g}}$, given by (4.29), is equal region $R_{\boldsymbol{g}}$, given by (2.54) ($\underline{a}_{ij} = \overline{a}_{ij} = a_{ij}$ $\forall i,j$). In this case, for all $\boldsymbol{x} \in R^u_{\boldsymbol{g}}$, inequality (4.32) can be expressed as $z_i(k) \succeq \bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\} = \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j(k-1)\} = \overline{a}_{ig_i} \otimes x_{g_i}(k-1)$. Therefore, it is straightforward to see that the set (4.33) is equal the set (2.60).*

5 Reachability Analysis of uMPL systems

In chapter 4, it was introduced a procedure to partition the state space of an uMPL system into components that can be completely characterized by DBM. In this chapter, this result is used to extend most of the results on reachability analysis, presented in (ADZKIYA *et al.*, 2015; ADZKIYA *et al.*, 2014b; ADZKIYA *et al.*, 2014a), to uMPL systems. The algorithms proposed have the same worst-case complexity as the corresponding for deterministic MPL systems. In the following, it is shown that the image and the inverse image of a set represented by a DBM through each subsystem of a partitioned uMPL system can be represented by a DBM, and therefore the DBM approach is useful for reachability analysis of uMPL systems.

Proposition 5.1 is an extension to uMPL systems of Proposition 3.1.

Proposition 5.1 *The image and the inverse image of a set represented by a DBM w.r.t. a subsystem of a partitioned uMPL system is a set that can be represented by a DBM.*

Proof:

The proof will be given for the image instance. The proof for the inverse image is similar. Each subsystem of a partitioned uMPL system can be represented by[1]*:*

$$x_i(k) \in \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\}, \quad \overline{a}_{ig_i} \otimes x_{g_i}(k-1)\right], \text{ if } \boldsymbol{x}(k-1) \in R^u_{\boldsymbol{g}},$$

where: $i \in \{1, ..., n\} \cup \{0\}$ and, for all $\boldsymbol{g}$, g_0 is set to 0, $a_{00} = 0$, $a_{0j} = \varepsilon$ for all $j \in \{1, ..., p\}$ and $a_{i0} = \varepsilon$ for all $i \in \{1, ..., n\}$.

Note that, given a set X_{k-1}, only the points in the intersection $X_{k-1} \cap R^u_{\boldsymbol{g}}$ are governed by this dynamics i.e.:

$$x_i(k) \in \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\}, \quad \overline{a}_{ig_i} \otimes x_{g_i}(k-1)\right] \ \forall i, \text{ if } \boldsymbol{x}(k-1) \in X_{k-1} \cap R^u_{\boldsymbol{g}}. \tag{5.1}$$

If X_{k-1} can be represented by a DBM, the intersection $X_{k-1} \cap R^u_{\boldsymbol{g}}$ can also be represented by a DBM that will be noted by $D^{(X_{k-1} \cap R^u_g)}$, with entries $d^{(X_{k-1} \cap R_g)}_{ir} = (\mathtt{d}^{(X_{k-1} \cap R_g)}_{ir}, \leq)$. Since computing the canonical form does not change the region represented by a DBM, it will be assumed that $D^{(X_{k-1} \cap R^u_g)}$ is in the canonical form.

[1] This model considers an additional equation corresponding to the artificial variable: $x_0 = 0 + x_0$

Therefore for all $\boldsymbol{x}(k-1) \in X_{k-1} \cap R^u_{\boldsymbol{g}}$ *we have that the tightest possible upper bound for* $x_i(k-1) - x_j(k-1)$ *is given by:*

$$x_i(k-1) - x_r(k-1) \leq \mathtt{d}_{ir}^{(X_{k-1} \cap R^u_g)}, \quad \forall i, r. \tag{5.2}$$

According to (5.1) *we have that*[2]*:*

$$\begin{aligned} x_i(k) - x_r(k) \ \ \in \ \ & \Bigg[\bigoplus_{j=1}^{p} \{\underline{a}_{ij} \otimes x_j(k-1)\} - \overline{a}_{rg_r} \otimes x_{g_r}(k-1), \\ & \qquad \overline{a}_{ig_i} \otimes x_{g_i}(k-1) - \bigoplus_{j=1}^{p} \{\underline{a}_{rj} \otimes x_r(k-1)\} \Bigg], \ \forall i, r. \end{aligned} \tag{5.3}$$

From (5.3) *we have that:*

$$\begin{cases} x_i(k) - x_r(k) \geq \bigoplus_{j=1}^{p} \{\underline{a}_{ij} \otimes x_j(k-1)\} - \overline{a}_{rg_r} \otimes x_{g_r}(k-1), \ \forall i, r, & (5.4) \\ x_i(k) - x_r(k) \leq \overline{a}_{ig_i} \otimes x_{g_i}(k-1) - \bigoplus_{j=1}^{p} \{\underline{a}_{rj} \otimes x_r(k-1)\}, \ \forall i, r. & (5.5) \end{cases}$$

Inequality (5.4) *can be expressed as:*

$$x_r(k) - x_i(k) \leq \min_j \{x_{g_r}(k-1) - x_j(k-1) + \overline{a}_{rg_r} - \underline{a}_{ij}\}, \ \forall i, r. \tag{5.6}$$

From (5.2) *we have that:*

$$x_r(k) - x_i(k) \leq \min_j \{\mathtt{d}_{g_r j}^{(X_{k-1} \cap R^u_g)} + \overline{a}_{rg_r} - \underline{a}_{ij}\}, \ \forall i, r. \tag{5.7}$$

Similarly, inequality (5.5) *can be expressed as:*

$$x_i(k) - x_r(k) \leq \min_j \{\mathtt{d}_{g_i j}^{(X_{k-1} \cap R^u_g)} + \overline{a}_{ig_i} - \underline{a}_{rj}\}, \ \forall i, r. \tag{5.8}$$

Inequalities (5.7) *and* (5.8) *define the same region. This can be checked by noticing that replacing i with r and r with i in* (5.8) *one obtains* (5.7)*. Therefore, inequalities* (5.4) *and* (5.5) *are completely represented by* (5.8)*. Thus, tightest possible upper bound for* $x_i(k) - x_j(k)$ *is given by:*

$$x_i(k) - x_r(k) \leq \min_j \{\mathtt{d}_{g_i j}^{(X_{k-1} \cap R^u_g)} + \overline{a}_{ig_i} - \underline{a}_{rj}\}, \ \forall i, r. \tag{5.9}$$

[2] From the interval analysis theory: $[\mathbf{x}] - [\mathbf{y}] = [\underline{x} - \overline{y}, \ \overline{x} - \underline{y}]$

Following the same arguments given in the proof for deterministic systems, note that all points in the image of X_{k-1} w.r.t. a subsystem $\boldsymbol{g}$ of a partitioned uMPL system must satisfy (5.9)*. Otherwise, at least one of the restrictions defined by the dynamics* (5.1) *would be violated. Moreover, all the points that satisfy* (5.9) *can be reached from $X_{k-1} \cap R^u_{\boldsymbol{g}}$. Thus, the image of X_{k-1} w.r.t. the subsystem $\boldsymbol{g}$ of a partitioned uMPL system is given by the region defined by* (5.9)*, which can be represented by a DBM $D^{(X_k|g)}$ with entries defined by:*

$$d^{(X_k|g)}_{ir} = (\min_j \{\mathsf{d}^{(X_{k-1}\cap R^u_g)}_{g_i j} + \overline{a}_{ig_i} - \underline{a}_{rj}\},\ \leq). \tag{5.10}$$

■

Given a DBM $D^{(X_{k-1})}$ representing a set X_{k-1}, Algorithm 5.1 computes the image of X_{k-1} w.r.t. a subsystem of the partitioned uMPL system.

Algorithm 5.1: Computing the image of a DBM w.r.t a subsystem of a partitioned uMPL system

---

input : $D^{(X_{k-1})} \in \mathcal{B}^{(p+1)\times(p+1)}$ `// a DBM representing a region` $X_{k-1} \in \mathbb{R}^p$.

: $D^{(\mathbf{g})} \in \mathcal{B}^{(n+p+1)\times(n+p+1)}$ `// a DBM representing a subsystem of a partitioned uMPL system generated by a matrix` $A \in \overline{\mathbb{R}}^{n\times p}_{max}$.

output: $D^{(X_k|\mathbf{g})} \in \mathcal{B}^{(n+1)\times(n+1)}$ `// a DBM representing the image of` X_{k-1} `w.r.t. the subsystem` $\mathbf{g}$ `of the partitioned system.`

1. $D^{(\mathbb{R}^n)} \leftarrow e_{\mathcal{B}^{n+1\times n+1}}$ `// a DBM representing` $\mathbb{R}^n$
2. $D^{(\mathbb{R}^n\times X_{k-1})} \leftarrow D^{(\mathbb{R}^n)} \times D^{(X_{k-1})}$ `// compute the cart. product (see section 2.4.2)`
3. $D^{(\bar{X}_k)} \leftarrow D^{(\mathbb{R}^n\times X_{k-1})} \oplus_{\mathcal{B}} D^{\mathbf{g}}$ `// compute the intersection (see remark 2.21).`
4. $D^{(\bar{X}_k)} \leftarrow cf(D^{(\bar{X}_k)})$ `// compute the canonical form (see section 2.4.1).`
5. $D^{(X_k|\mathbf{g})} \leftarrow D^{(\bar{X}_k)}\lceil_{x'_1,\ldots,x'_n}$ `// compute the orthogonal projection over` $\mathbf{x}(k)$ `(see section 2.4.2).`

In the following, is a discussion on how Algorithm 5.1 yields the region defined (5.10), which represents the image of a set X_{k-1} w.r.t. a subsystem $\mathbf{g}$ of the partitioned uMPL system. Note that, the DBM $D^{(\bar{X}_k)}$ obtained in step 3 of Algorithm 5.1 exactly represents (5.1). Moreover, by definition, the DBM obtained in step 4 (which is the canonical form representation of $D^{(\bar{X}_k)}$) has the tightest possible bounds. Therefore, the DBM $D^{(X_k|\mathbf{g})}$, obtained in the step 5 as orthogonal projection of the canonical form over the variables $\mathbf{x}(k)$, is the DBM defined by (5.10).

Similarly, given a DBM $D^{(X_{-k+1})}$ representing a set X_{-k+1}, Algorithm 5.2 computes the inverse image of X_{-k+1} w.r.t. a subsystem of the partitioned uMPL system.

Algorithm 5.2: Computing the inverse image of a DBM w.r.t a subsystem of a partitioned uMPL system

input : $D^{(X_{-k+1})} \in \mathcal{B}^{(n+1)\times(n+1)}$ `// a DBM representing a region` $X_{-k+1} \in \mathbb{R}^n$.
: $D^{(\mathbf{g})} \in \mathcal{B}^{(n+p+1)\times(n+p+1)}$ `// a DBM representing a subsystem of a partitioned uMPL system generated by a matrix` $A \in \overline{\mathbb{R}}_{max}^{n\times p}$.

output: $D^{(X_{-k}|\mathbf{g})} \in \mathcal{B}^{(p+1)\times(p+1)}$ `// a DBM representing the inverse image of` X_{-k+1} `w.r.t. the subsystem` **g** `of the partitioned system.`

1 $D^{(\mathbb{R}^p)} \leftarrow e_{\mathcal{B}^{p+1\times p+1}}$ `// a DBM representing` $\mathbb{R}^n$
2 $D^{(X_{-k+1}\times\mathbb{R}^p)} \leftarrow D^{(X_{-k+1})} \times D^{(\mathbb{R}^p)}$ `// compute the cart. product (see section 2.4.2)`
3 $D^{(\bar{X}_{-k})} \leftarrow D^{(X_{-k+1}\times\mathbb{R}^p)} \oplus_{\mathcal{B}} D^{\mathbf{g}}$ `// compute the intersection (see remark 2.21).`
4 $D^{(\bar{X}_{-k})} \leftarrow cf(D^{(\bar{X}_{-k})})$ `// compute the canonical form (see section 2.4.1).`
5 $D^{(X_{-k}|\mathbf{g})} \leftarrow D^{(\bar{X}_{-k})}\lceil_{x_1,\ldots,x_n}$ `// compute the orthogonal projection over` $\mathbf{x}(k-1)$ `(see section 2.4.2).`

The worst-case complexity of Algorithms 5.1 and 5.2 critically depends on computing the canonical form representation of a DBM in $\mathcal{B}^{(n+p+1)\times(n+p+1)}$ (step 4 for both algorithms), which has cubic complexity w.r.t its dimensions. Thus, the worst-case complexity is $\mathcal{O}((n+p)^3)$.

Corollary 5.2 *The image of a set represented by union of finitely many DBM w.r.t. a partitioned uMPL system can be represented by union of finitely many DBM.*

Given a partitioned uMPL system generated by a matrix $A \in \overline{\mathbb{R}}_{max}^{n\times p}$, computing the image (or the inverse image) of a union of q DBM can be done by computing the image (or the inverse image) of each DBM w.r.t each subsystem of the partitioned uMPL system. Thus the worst-case complexity depends on the number of DBM (considered to be q), on the worst-case cardinality of the collection of subsystem, given by p^n, and on the complexity of computing the image (or the inverse image) of each DBM w.r.t. each subsystem of a partitioned uMPL system, which is $\mathcal{O}((n+p)^3)$. Therefore, the worst-case complexity is $\mathcal{O}(qp^n(n+p)^3)$.

Remark 5.3 *For autonomous uMPL systems, parameter p equals n, and therefore the worst-case complexity of computing the image (or the inverse image) of q DBM w.r.t the system is $\mathcal{O}(qn^{n+3})$. For nonautonomous uMPL systems, parameter p equals $n+m$, and therefore the worst-case complexity is $\mathcal{O}(q(n+m)^{n+3})$. Note that this is the same worst case complexity of computing the image (or the inverse image) of q DBM w.r.t a PWA system generated by an MPL system (see Remark 3.3).*

Note that, the procedures for computing the image and the inverse image of a DBM w.r.t a subsystem of a partitioned uMPL system (Algorithms 5.1 and 5.2, respectively) have, essentially, the same steps of the procedures for computing the image and the inverse image of a DBM w.r.t a subsystem of a PWA system generated by an MPL system (Algorithms 3.1 and 3.2, respectively). Consequently, as presented in the following sections, forward and backward reachability analysis of uMPL systems can be performed by using a procedure that is quite similar to the procedures presented in sections 3.1 and 3.2.

In the following sections, it will be assumed that the set of initial/final conditions $X_0 \subseteq \mathbb{R}^n$ and the set of control $U_k \subseteq \mathbb{R}^m$, at each event step, are a union of q_0 and r_k DBM, respectively. Moreover, the cardinality of the DBM union set representing X_k at event step k will be noted by q_k.

5.1 Forward Reachability Analysis

Similarly to the classical case presented in chapter 3, the forward reachability analysis of uMPL systems concerns the computation of the set of all states that may be reached from a set of initial states via the uMPL dynamics, at a particular event step (the reach set) or over a set of consecutive events (reach tube). In the following, we recall the definitions of reach set and reach tube.

Definition 5.4 (reach set) *Given an uMPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach set** X_N at the event step $N > 0$ is the set of all states $\{\boldsymbol{x}(N) : \boldsymbol{x}(0) \in X_0\}$ that can be reached via the uMPL dynamics, possibly by application of controls.*

Definition 5.5 (reach tube) *Given an uMPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach tube** is defined by the set-valued function $k \mapsto X_k$ for any given $k > 0$ where X_k is defined.*

5.1.1 Forward Reachability Analysis of Autonomous uMPL systems

Given an *autonomous* uMPL system and a nonempty set of initial conditions X_0, the reach set X_k at the event step k can be recursively calculated as the image of the reach set X_{k-1} w.r.t the uMPL dynamics:

$$X_k = \mathcal{I}_{[\mathbf{A}]}\{X_{k-1}\} = \{A \otimes \mathbf{x} : \mathbf{x} \in X_{k-1},\ A \in [\mathbf{A}]\} = [\mathbf{A}] \otimes X_{k-1}. \tag{5.11}$$

From Corollary 5.2, if X_{k-1} can be represented by a union of q_{k-1} DBM, then $X_k = \mathcal{I}_{[\mathbf{A}]}\{X_{k-1}\}$ can be represented by a union of q_k DBM. Thus, by induction, it can be concluded that if X_0 can be represented by a union of q_0 DBM, then X_k can be represented by a union of q_k DBM, for each $k \in \mathbb{N}$.

Given the set of initial conditions X_0, computing the reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the partitioned uMPL system generated by $[\mathbf{A}]$; then, for each $k \in \{1, ..., N\}$, compute the image of X_{k-1} w.r.t. the partitioned uMPL system. The worst-case complexity to compute $\mathcal{I}_{[\mathbf{A}]}\{X_{k-1}\}$, for each $k \in \{1, ..., N\}$ is $\mathcal{O}(q_{k-1}n^{n+3})$ (see remark 5.3). Thus, the overall complexity is $\mathcal{O}(n^{n+3}\sum_{k=1}^{N} q_{k-1})$.

Remark 5.6 *As in the deterministic case, in general , it is not possible to quantify the exact cardinality q_k of the DBM union set at event step k a priori (see remark 3.6). The worst-case cardinality depends on the cardinality of the DBM union set at event step $k-1$, given by q_{k-1}, and on the worst-case cardinality of the number of regions of the partitioned uMPL system, given by n^n. Therefore the worst-case cardinality is $q_{k-1}n^n$. In practice, many regions and intersections of DBM and regions are empty, then the cardinality q_k is drastically smaller than its worst-case bound.*

In the following, we extend the one-shot procedure presented in section 3.1.1 to uMPL systems. Given a nonempty set of initial conditions X_0, the reach set X_N at the event step N can be computed, in a one-shot procedure, by using the following formula:

$$X_N = \mathcal{I}_{[\mathbf{A}]^{\otimes N}}\{X_0\} = \{\mathcal{A} \otimes \mathbf{x} : \mathbf{x} \in X_0,\ \mathcal{A} \in [\mathbf{A}]^{\otimes N}\} = [\mathbf{A}]^{\otimes N} \otimes X_0. \tag{5.12}$$

A general procedure for computing X_N is: 1) compute $[\mathbf{A}]^{\otimes N}$ (see (4.16)); then, 2) construct the partitioned uMPL system generated by $[\mathbf{A}]^{\otimes N}$; and, 3) compute the image of X_0 w.r.t. the obtained partitioned system. The complexity of this procedure is $\mathcal{O}(\lceil \log_2(N) \rceil n^3 + q_0 N^3)$, the same as the one-shot procedure presented in section 3.1.1.

Example 5.7 *Consider the autonomous uMPL system given by:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1),\ \textit{where}\ A(k) \in \begin{pmatrix} [4,\ 6] & [3,\ 5] \\ [3,\ 7] & [4,\ 5] \end{pmatrix}.$$

In example 4.11 this system was represented as a collection of DBM $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}\}$.

Given $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \le x_1 \le 1,\ 1 \le x_2 \le 3\}$, the reach sets X_k for $k \in \{1, 2\}$ are computed in the following. Note that the set X_0 can be represented by the following DBM:

$$D^{(X_0)} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) \\ (1,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array}\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

To compute the reach set $X_1 = \mathcal{I}_{[\boldsymbol{A}]}\{X_0\}$, we must compute the image of X_0 w.r.t each component $\boldsymbol{g}$ of the partitioned uMPL system. According to algorithm 5.1, the image of $D^{(X_0)}$ w.r.t. the component $\boldsymbol{g} = (1,1)$ can be computed as follows: first, we compute the Cartesian product of $D^{(\mathbb{R}^2)}$ (a DBM representing $\mathbb{R}^2$) and $D^{(X_0)}$:

$$D^{(\mathbb{R}^2 \times X_0)} = D^{(\mathbb{R}^2)} \times D^{(X_0)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array}\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

Then, we compute the intersection of $D^{(\mathbb{R}^2 \times X_0)}$ and $D^{(1,1)}$:

$$D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (6,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (7,\leq) & \varepsilon_{\mathcal{B}} \\ (1,\leq) & (-4,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3,\leq) & (-3,\leq) & (-4,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array}\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

Next, we compute the canonical form representation of the intersection:

$$cf(D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)}) = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-4,\leq) & (-5,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ (7,\leq) & e_{\mathcal{B}} & (2,\leq) & (6,\leq) & (6,\leq) \\ (8,\leq) & (3,\leq) & e_{\mathcal{B}} & (7,\leq) & (7,\leq) \\ (1,\leq) & (-4,\leq) & (-4,\leq) & e_{\mathcal{B}} & e_{\mathcal{B}} \\ (2,\leq) & (-3,\leq) & (-4,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array}\begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

Finally, we compute the orthogonal projection of the canonical form over the variables x_1' and x_2'. The image of X_0 w.r.t the component $\boldsymbol{g} = (1,1)$ is noted by $X_1|_{\boldsymbol{g}=(1,1)}$ and represented by the following DBM.

$$D^{(X_1|_{g=(1,1)})} = cf(D^{(\mathbb{R}^2 \times X_0)} \oplus_{\mathcal{B}} D^{(1,1)})\lceil_{\boldsymbol{x}'} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-4, \leq) & (-5, \leq) \\ (7, \leq) & e_{\mathcal{B}} & (2, \leq) \\ (8, \leq) & (3, \leq) & e_{\mathcal{B}} \end{array} \right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}$$

Applying the same procedure for $D^{(2,1)}$ and $D^{(2,2)}$ we obtain:

$$D^{(X_1|_{g=(2,1)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-4, \leq) & (-5, \leq) \\ (8, \leq) & e_{\mathcal{B}} & (1, \leq) \\ (8, \leq) & (3, \leq) & e_{\mathcal{B}} \end{array} \right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}$$

$$D^{(X_1|_{g=(2,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-5, \leq) & (-6, \leq) \\ (8, \leq) & e_{\mathcal{B}} & (1, \leq) \\ (8, \leq) & (2, \leq) & e_{\mathcal{B}} \end{array} \right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array}$$

The reach set X_1 is the union of the images of X_0 w.r.t. each component of partitioned uMPL system and it is represented by the collection of DBM given by $\mathtt{D}^{(X_1)} = \{D^{(X_1|_{g=(1,1)})}, D^{(X_1|_{g=(2,1)})}, D^{(X_1|_{g=(2,2)})}\}$. However, note that, $D^{(X_1|_{g=(2,1)})} \oplus_{\mathcal{B}} D^{(X_1|_{g=(2,2)})} = D^{(X_1|_{g=(2,2)})}$, and therefore, according to remark 2.22, $D^{(X_1|_{g=(2,1)})} \cup D^{(X_1|_{g=(2,2)})} = D^{(X_1|_{g=(2,1)})}$. Then, the DBM union set can be simplified to $\mathtt{D}^{(X_1)} = \{D^{(X_1|_{g=(1,1)})}, D^{(X_1|_{g=(2,1)})}\}$. Therefore, $X_1 = \mathcal{R}(D^{(X_1|_{g=(2,1)})}) \cup \mathcal{R}(D^{(X_1|_{g=(2,1)})}) = \{\boldsymbol{x}' \in \mathbb{R}^2 : 4 \leq x_1' \leq 7,\ 5 \leq x_2' \leq 8,\ -2 \leq x_2' - x_1' \leq 3\} \cup \{\boldsymbol{x}' \in \mathbb{R}^2 : 4 \leq x_1' \leq 8,\ 5 \leq x_2' \leq 8,\ -1 \leq x_2' - x_1' \leq 3\}$

The reach set X_2 is obtained by computing the image of each DBM in $\mathtt{D}^{(X_1)}$ w.r.t each DBM in $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}\}$, which yields $X_2 = \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \leq x_1' \leq 14,\ 9 \leq x_2' \leq 15,\ -3 \leq x_2' - x_1' \leq 3\}$. The reach sets X_1 and X_2 are shown in Figure 9.

Remark 5.8 *In general, the uMPL systems are expansive in the sense that, given X_0, the hyper-volume[3] of the reach sets X_k tends to increase with k (see Figure 9 for instance).*

In the following it is shown that under specific conditions the structure of the uMPL dynamics leads to savings for the computation of the reach tube. Consider a matrix of intervals $[\mathbf{A}] = [\underline{A},\ \overline{A}]$ such that: $\underline{A} \in \overline{\mathbb{R}}^{n \times n}_{max}$ is an irreducible matrix with cyclicity c_1 and

[3] the hyper-volume of a set $X_k \in \mathbb{R}^n$ is given by $V = \int_{X_k} d\mathbf{x}$

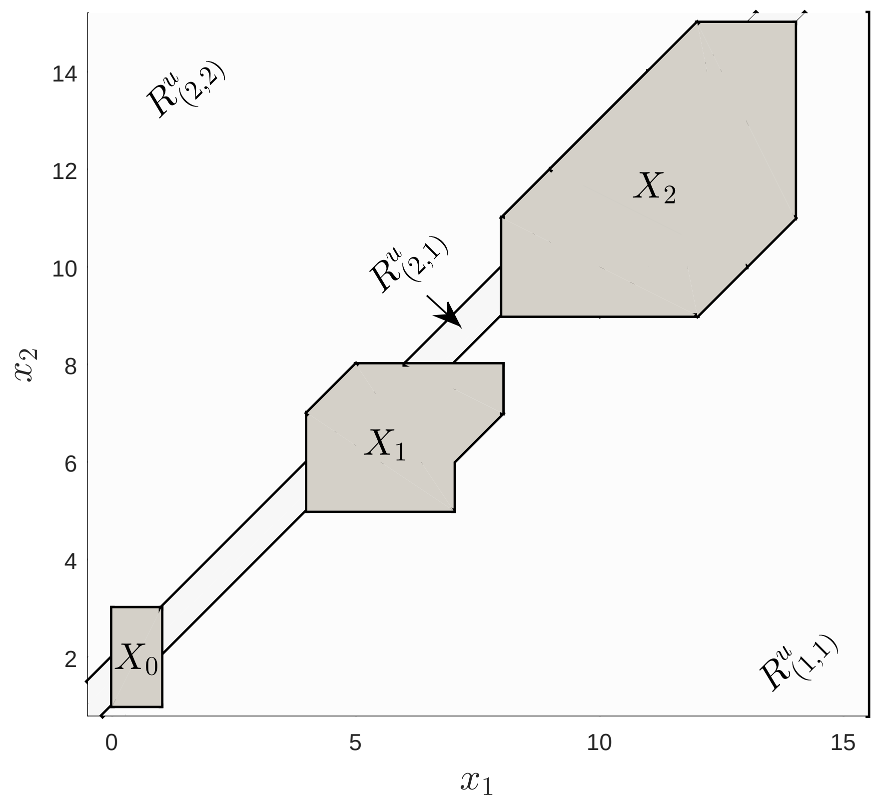

Figure 9 – reach tube for $k \in \{1,2\}$ (autonomous uMPL system).

max-plus eigenvalue λ_1; and $\overline{A} \in \overline{\mathbb{R}}^{n\times n}_{max}$ is an irreducible matrix with cyclicity c_2 and max-plus eigenvalue λ_2. From equation (4.16), we have that:

$$[\mathbf{A}]^{\otimes k} = [\underline{A}^{\otimes k},\ \overline{A}^{\otimes k}]. \tag{5.13}$$

From Proposition 2.13, there exist integers $K_0(\underline{A})$ and $K_0(\overline{A})$ such that:

$$k \geq K_0(\underline{A}) \quad \Rightarrow \quad \underline{A}^{\otimes(k+c_1)} = \lambda_1^{\otimes c_1} \otimes \underline{A}^{\otimes k}, \tag{5.14}$$

$$k \geq K_0(\overline{A}) \quad \Rightarrow \quad \overline{A}^{\otimes(k+c_2)} = \lambda_2^{\otimes c_2} \otimes \overline{A}^{\otimes k}. \tag{5.15}$$

In the special case where $c_1 = c_2$ and $\lambda_1 = \lambda_2$, there exists an integer $K_0(\underline{A}, \overline{A}) = \max\{K_0(\underline{A}), K_0(\overline{A})\}$ such that:

$$\begin{aligned} k \geq K_0(\underline{A}, \overline{A}) \Rightarrow [\mathbf{A}]^{\otimes(k+c_1)} &= [\underline{A}^{\otimes(k+c_1)},\ \overline{A}^{\otimes(k+c_1)}] \\ &= [\lambda_1^{\otimes c_1} \otimes \underline{A}^{\otimes k},\ \lambda_1^{\otimes c_1} \otimes \overline{A}^{\otimes k}] \\ &= \lambda_1^{\otimes c_1} \otimes [\underline{A}^{\otimes k},\ \overline{A}^{\otimes k}] \\ &= \lambda_1^{\otimes c_1} \otimes [\mathbf{A}]^{\otimes k}. \end{aligned} \tag{5.16}$$

Therefore, in this special case, given a set of initial positions X_0 there exists $k_0(X_0) = max\{k_0(x)\}$ such that,

$$k \geq k_0(X_0) \Rightarrow X_{k+c_1} = \lambda_1^{c_1} \otimes X_k. \tag{5.17}$$

Thus, in order do compute X_N, $N > k_0(X_0) + c - 1$, it is only necessary to compute $X_1, ...,$ $X_{k_0(X_0)+c-1}$.

Example 5.9 *Consider the uMPL system characterized by the following matrix of intervals:*

$$[\boldsymbol{A}] = \begin{pmatrix} [0,\ 2] & 5 \\ 3 & [0,\ 3] \end{pmatrix}.$$

or equivalently,

$$[\boldsymbol{A}] = [\underline{A},\ \overline{A}],\ \textit{where: } \underline{A} = \begin{pmatrix} 0 & 5 \\ 3 & 0 \end{pmatrix} \textit{ and } \overline{A} = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix}.$$

Both matrices, $\underline{A}$ and $\overline{A}$, have cyclicity $c = 2$ and max-plus eigenvalue $\lambda = 4$ (see section 2.3). Moreover, as can be observed in Figure 10, given $X_0 = \{\boldsymbol{x} \in \mathbb{R}^2 : 0 \leq x_1 \leq 1,\ 1 \leq x_2 \leq 3\}$, for all $k \geq 1$ we have that $X_{k+2} = 4^{\otimes 2} \otimes X_k = 8 \otimes X_k$.

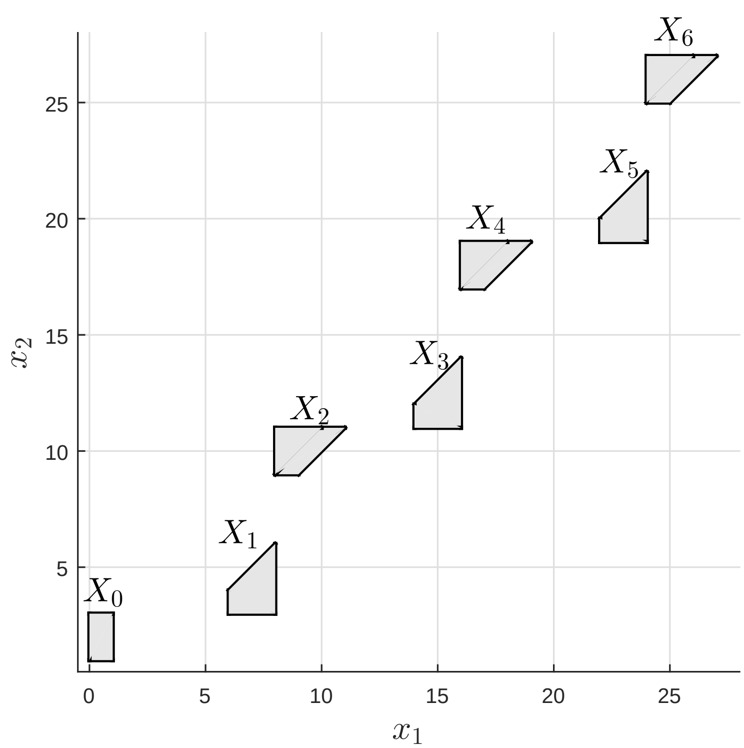

Figure 10 – cyclic behavior of an uMPL system.

Remark 5.10 *The column space or image of a matrix of intervals $[\boldsymbol{A}] \in \overline{\mathbb{R}}_{max}^{n\times p}$ can be defined as $Im\,[\boldsymbol{A}] = \{\boldsymbol{x'} = A\otimes\boldsymbol{x} : \boldsymbol{x}\in\mathbb{R}^p,\ A\in[\boldsymbol{A}]\}$. Note that, $Im\,[\boldsymbol{A}]$ can be computed as the image of $\overline{\mathbb{R}}_{max}^{p}$ w.r.t. the partitioned uMPL system generated by $[\boldsymbol{A}]$. According to algorithm 5.1, the image of $\overline{\mathbb{R}}_{max}^{p}$ w.r.t. each subsystem of the partitioned uMPL system can be calculated by computing the DBM $D^{(\mathbb{R}^n\times\mathbb{R}^p)} = D^{(\mathbb{R}^{n+p})}$, which represents $\mathbb{R}^n\times\mathbb{R}^p$ (step 2); then computing $cf(D^{(\mathbb{R}^{n+p})}\oplus_{\mathcal{B}} D^{(\boldsymbol{g})})$ (steps 3 and 4); and finally projecting the canonical form over $\boldsymbol{x'}$ (step 5). However, note that $D^{(\mathbb{R}^{n+p})}\oplus_{\mathcal{B}} D^{(\boldsymbol{g})} = D^{(\boldsymbol{g})}$, then the image of $[\boldsymbol{A}]$ can be computed by computing the canonical form of each DBM representing the the partitioned uMPL system generated by $[\boldsymbol{A}]$ and then projecting the canonical form over $\boldsymbol{x'}$.*

Example 5.11 *Consider the matrix*

$$[\boldsymbol{A}] = \begin{pmatrix} [4,\ 6] & [3,\ 5] \\ [3,\ 7] & [4,\ 5] \end{pmatrix}.$$

The partitioned uMPL system generated by this matrix is represented by the collection of DBM $\mathtt{D} = \{D^{(1,1)},\ D^{(2,1)},\ D^{(2,2)}\}$, computed in example 4.11.

The image of $[\boldsymbol{A}]$ is computed as follows: First we compute the canonical form of the DBM $D^{(1,1)}$, $D^{(2,1)}$ and $D^{(2,2)}$:

$$cf(D^{(1,1)}) = \begin{array}{c} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (3,\leq) & (6,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} & (7,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array},$$

$$cf(D^{(2,1)}) = \begin{array}{c} \begin{array}{ccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (1,\leq) & (7,\leq) & (5,\leq) \\ \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} & (7,\leq) & (6,\leq) \\ \varepsilon_{\mathcal{B}} & (-4,\leq) & (-5,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & (2,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \end{array},$$

$$cf(D^{(2,2)}) = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (1,\leq) & \varepsilon_{\mathcal{B}} & (5,\leq) \\ \varepsilon_{\mathcal{B}} & (2,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) \\ \varepsilon_{\mathcal{B}} & (-5,\leq) & (-6,\leq) & e_{\mathcal{B}} & (-2,\leq) \\ \varepsilon_{\mathcal{B}} & (-3,\leq) & (-4,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}.$$

Then, we project the canonical form over the variables $\boldsymbol{x}'$*:*

$$cf(D^{(1,1)})\lceil_{\boldsymbol{x}'} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (3,\leq) \\ \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array},$$

$$cf(D^{(2,1)})\lceil_{\boldsymbol{x}'} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (1,\leq) \\ \varepsilon_{\mathcal{B}} & (3,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \end{array},$$

$$cf(D^{(2,2)})\lceil_{\boldsymbol{x}'} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (1,\leq) \\ \varepsilon_{\mathcal{B}} & (2,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}.$$

The image of $[\boldsymbol{A}]$ *can be represented by* $\mathtt{D}^{(Im[\boldsymbol{A}])} = \{cf(D^{(1,1)})\lceil_{\boldsymbol{x}'},\ cf(D^{(2,1)})\lceil_{\boldsymbol{x}'},\ cf(D^{(2,2)})\lceil_{\boldsymbol{x}'}\}$. *However, note that,* $cf(D^{(1,1)})\lceil_{\boldsymbol{x}'} \oplus_{\mathcal{B}} cf(D^{(2,1)})\lceil_{\boldsymbol{x}'} = cf(D^{(2,1)})\lceil_{\boldsymbol{x}'}$ *and* $cf(D^{(1,1)})\lceil_{\boldsymbol{x}'} \oplus_{\mathcal{B}} cf(D^{(2,2)})\lceil_{\boldsymbol{x}'} = cf(D^{(2,2)})\lceil_{\boldsymbol{x}'}$*. Thus,* $\mathtt{D}^{(Im[\boldsymbol{A}])} = \{cf(D^{(1,1)})\lceil_{\boldsymbol{x}'}\}$ *(see remark 2.22). Then,* $Im\,[\boldsymbol{A}] = \{\boldsymbol{x}' \in \mathbb{R}^2 : -3 \leq x_2 - x_1 \leq 3\}$.

5.1.2 Forward Reachability Analysis of Nonautonomous uMPL systems

For *nonautonomous* uMPL systems, forward reachability analysis can be performed by first representing the systems as an augmented autonomous uMPL system (see equation (4.3)), then given a nonempty set of initial conditions X_0 and the set of inputs U_k for each $k \in \mathbb{N}$, the reach set X_k at the event step k can be recursively calculated as:

$$X_k = \mathcal{I}_{[\mathbf{F}]}\{X_{k-1} \times U_k\} = \{F \otimes \mathbf{y} : \mathbf{y} \in X_{k-1} \times U_k,\ F \in [\mathbf{F}]\}. \tag{5.18}$$

where $[\mathbf{F}] = ([\mathbf{A}]\ \ [\mathbf{B}])$ and $\mathbf{y} = \left(\mathbf{x}^T\ \mathbf{u}^T\right)^T$.

If X_{k-1} and U_k can be represented by a union of q_{k-1} and r_k DBM , respectively, then $X_{k-1} \times U_k$ can be represented by a union of $\bar{q}_{k-1} = q_{k-1} r_k$ DBM. Thus, from Corollary 5.2, $X_k = \mathcal{I}_{[\mathbf{F}]}\{X_{k-1} \times U_k\}$ can be represented by a union of q_k DBM. By induction, it can be concluded that if X_0 can be represented by a union of q_0 DBM and U_k can be represented by a union of r_k DBM for each $k \in \mathbb{N}$, then X_k can be represented by a union of q_k DBM, for each $k \in \mathbb{N}$.

Given a nonautonomous uMPL system, the set of initial conditions X_0 and set of inputs U_k for each $k \in \{1, ..., N\}$, computing the reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the partitioned uMPL system generated by $[\mathbf{F}] = ([\mathbf{A}] \ [\mathbf{B}])$; then, for each $k \in \{1, ..., N\}$, compute the image of $X_{k-1} \times U_k$ w.r.t. the partitioned uMPL system. The worst-case complexity to compute $\mathcal{I}_F\{X_{k-1} \times U_k\}$, for each $k \in \mathbb{N}$ is $\mathcal{O}(\bar{q}_{k-1}(n+m)^{n+3})$ (see remark 5.3). Thus, the overall complexity is $\mathcal{O}((n+m)^{n+3} \sum_{k=1}^{N} \bar{q}_{k-1})$.

The set of all states that can be reached in N event steps can be computed using a one-shot procedure. Given a nonempty set of initial conditions X_0, the reach set X_N at the event step N is given by:

$$X_N = ([\mathbf{A}]^{\otimes N},\ [\mathbf{A}]^{\otimes (N-1)} \otimes [\mathbf{B}], ..., [\mathbf{B}]) \otimes (X_0 \times U_1 \times ... \times U_N). \tag{5.19}$$

Given the matrices $[\mathbf{A}]$ and $[\mathbf{B}]$, a set of initial conditions X_0 (represented by a union of q_0 DBM) and a sequence of input sets $U_1, ..., U_N$, a general procedure for computing X_N is given by: 1) generate the matrix $([\mathbf{A}]^{\otimes N},\ [\mathbf{A}]^{\otimes (N-1)} \otimes [\mathbf{B}], ..., [\mathbf{B}])$; then, 2) Construct the partitioned uMPL system generated by this matrix; and, 3) compute the image of $X_0 \times U_1 \times ... \times U_N$ w.r.t the obtained partitioned system. The complexity of steps 1, 2 and 3 is, respectively, $\mathcal{O}(Nn^3 + Nn^2m)$, $\mathcal{O}((n+mN)^{n+3})$ and $\mathcal{O}(q_0(n+mN)^{n+3})$. Note that , this approach is not tractable for problems over long event horizons, since the maximum number of regions of the partitioned uMPL system is $(n+mN)^n$ and grows polynomially w.r.t. the event horizon N.

Example 5.12 *Consider the nonautonomous uMPL system given by:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1) \oplus B(k) \otimes \boldsymbol{u}(k),$$

where,

$$A(k) \in \begin{pmatrix} 2 & [2,\ 4] \\ [3,\ 5] & [3,\ 4] \end{pmatrix} \text{ and } B(k) \in \begin{pmatrix} e \\ \varepsilon \end{pmatrix}.$$

In example 4.12 this system was represented as a collection of DBM $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}, D^{(3,1)}, D^{(3,2)}\}$.

Given $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 3\}$*, and the sequence of controls* $u_1(1) = 2.5$ *and* $u_1(2) = 8$*, the reach sets* X_k *for* $k \in \{1, 2\}$ *are computed in the following.*

The set of initial positions X_0 *and the control input* $u_1(1)$ *can be represented by the following DBM:*

$$D^{(X_0)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} \\ (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array} \qquad D^{(U_1)} = \begin{array}{c} \begin{array}{cc} x_0 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-2.5, \leq) \\ (2.5, \leq) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ u_1 \end{array} \end{array}$$

To compute the reach set $X_1 = \mathcal{I}_{[\boldsymbol{F}]}\{X_0 \times U_1\}$*, we must to compute the image of* $X_0 \times U_1$ *w.r.t each component* $\boldsymbol{g} \in \{D^{(1,1)},\ D^{(2,1)},\ D^{(2,2)},\ D^{(3,1)},\ D^{(3,2)}\}$ *of the partitioned uMPL system generated by* $[\boldsymbol{F}]$*. The Cartesian product* $X_0 \times U_1$ *can be represented by:*

$$D^{(X_0 \times U_1)} = \begin{array}{c} \begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5, \leq) \\ (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2.5, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \end{array} \end{array}$$

According to algorithm 5.1, the image of $D^{(X_0 \times U_1)}$ *w.r.t. the component* $\boldsymbol{g} = (1, 1)$ *can be computed as follows: first, we compute the Cartesian product of* $D^{(\mathbb{R}^2)}$ *(a DBM representing* $\mathbb{R}^2$*) and* $D^{(X_0 \times U_1)}$*:*

$$D^{(\mathbb{R}^2 \times X_0 \times U_1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x'_1 & x'_2 & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5, \leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2.5, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x'_1 \\ x'_2 \\ x_1 \\ x_2 \\ u_1 \end{array} \end{array}$$

Then, we compute the intersection of $D^{(\mathbb{R}^2 \times X_0 \times U_1)}$ *and* $D^{(1,1)}$*:*

$$D^{(\mathbb{R}^2\times X_0\times U_1)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \left(\begin{array}{cccccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1,\leq) & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3,\leq) & (-2,\leq) & (-3,\leq) & (-2,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2.5,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

Next, we compute the canonical form of the intersection:

$$cf(D^{(\mathbb{R}^2\times X_0\times U_1)} \oplus_{\mathcal{B}} D^{(1,1)}) = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \left(\begin{array}{cccccc} \top_{\mathcal{B}} & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5,\leq) \\ (3,\leq) & e_{\mathcal{B}} & (-1,\leq) & (2,\leq) & (3,\leq) & (0.5,\leq) \\ (6,\leq) & (3,\leq) & e_{\mathcal{B}} & (5,\leq) & (6,\leq) & (3.5,\leq) \\ (1,\leq) & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & (1,\leq) & (-1.5,\leq) \\ (-1,\leq) & (-4,\leq) & (-5,\leq) & (-2,\leq) & (-1,\leq) & (-3.,\leq) \\ (2.5,\leq) & e_{\mathcal{B}} & (-1,\leq) & (2,\leq) & (2.5,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

Note that, this is an empty DBM due to the fact that the set $X_0 \times U_1$ is not intersected with region $R^u_{(1,1)}$ (see example 4.9). Therefore, the image of X_0 w.r.t the component $\boldsymbol{g} = (1,1)$ is empty.

Now, let us compute the image of X_0 w.r.t the component $\boldsymbol{g} = (2,1)$. The intersection of $D^{(\mathbb{R}^2\times X_0\times U_1)}$ and $D^{(2,1)}$ is given by:

$$D^{(\mathbb{R}^2\times X_0\times U_1)} \oplus_{\mathcal{B}} D^{(2,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \left(\begin{array}{cccccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5,\leq) \\ \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5,\leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1,\leq) & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} \\ (3,\leq) & (-2,\leq) & (-3,\leq) & (1,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (2.5,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (5,\leq) & (4,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} \\ x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

The canonical form of the intersection is:

$$cf(D^{(\mathbb{R}^2\times X_0\times U_1)} \oplus_{\mathcal{B}} D^{(1,1)}) =$$
$$\begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-2.5,\le) & (-3,\le) & e_{\mathcal{B}} & e_{\mathcal{B}} & (-2.5,\le) \\ (6,\le) & e_{\mathcal{B}} & (1,\le) & (5,\le) & (4,\le) & (3.5,\le) \\ (6,\le) & (3,\le) & e_{\mathcal{B}} & (5,\le) & (6,\le) & (3.5,\le) \\ (1,\le) & (-2,\le) & (-3,\le) & e_{\mathcal{B}} & (1,\le) & (-1.5,\le) \\ (2,\le) & (-2,\le) & (-3,\le) & (1,\le) & e_{\mathcal{B}} & (-0.5,\le) \\ (2.5,\le) & e_{\mathcal{B}} & (-5,\le) & (2.5,\le) & (2.5,\le) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array} \end{array}$$

The image of X_0 w.r.t the component $\boldsymbol{g} = (2,1)$, noted by $X_1|_{\boldsymbol{g}=(2,1)}$, is given by the orthogonal projection of the canonical form over the variables x_1' and x_2', which is given by:

$$D^{(X_1|_{\boldsymbol{g}=(2,1)})} = cf(D^{(\mathbb{R}^2\times X_0\times U_1)} \oplus_{\mathcal{B}} D^{(1,1)})\lceil_{\boldsymbol{x}'} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-2.5,\le) & (-3,\le) \\ (6,\le) & e_{\mathcal{B}} & (1,\le) \\ (6,\le) & (3,\le) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array} \end{array}$$

The image of X_0 w.r.t the component $\boldsymbol{g} = (2,2)$ can be computed by applying the same procedure, which yields:

$$D^{(X_1|_{\boldsymbol{g}=(2,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-3,\le) & (-4,\le) \\ (7,\le) & e_{\mathcal{B}} & (1,\le) \\ (7,\le) & (2,\le) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array} \end{array}$$

The set $X_0 \times U_1$ is not intersected with the regions $R^u_{(3,1)}$ and $R^u_{(3,2)}$. Therefore, the image of X_0 w.r.t these components is the empty set.

Thus, the image of X_0 w.r.t uMPL system is represented by $\mathtt{D}^{(X_1)} = \{D^{(X_1|_{\boldsymbol{g}=(2,1)})}, D^{(X_1|_{\boldsymbol{g}=(2,2)})}\}$. Therefore, we have that $X_1 = \mathcal{R}(D^{(X_1|_{\boldsymbol{g}=(2,1)})}) \cup \mathcal{R}(D^{(X_1|_{\boldsymbol{g}=(2,2)})}) = \{\boldsymbol{x}' \in \mathbb{R}^2 : 2.5 \le x_1' \le 6,\ 3 \le x_2' \le 6,\ -1 \le x_2' - x_1' \le 3\} \cup \{\boldsymbol{x}' \in \mathbb{R}^2 : 3 \le x_1' \le 7,\ 4 \le x_2' \le 7,\ -1 \le x_2' - x_1' \le 2\}$.

The reach set X_2 is obtained by computing the image of each DBM in $\mathtt{D}^{(X_1)}$ w.r.t each DBM in $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}, D^{(3,1)}, D^{(3,2)}\}$, which yields $X_2 = \{\boldsymbol{x}' \in \mathbb{R}^2 : x_1' = 8,\ 6 \le x_2' \le 10\} \cup \{\boldsymbol{x}' \in \mathbb{R}^2 : 8 \le x_1' \le 11,\ 7 \le x_2' \le 12,\ -1 \le x_2' - x_1' \le 3\}$. The reach sets X_1 and X_2 are shown in Figure 11.

,

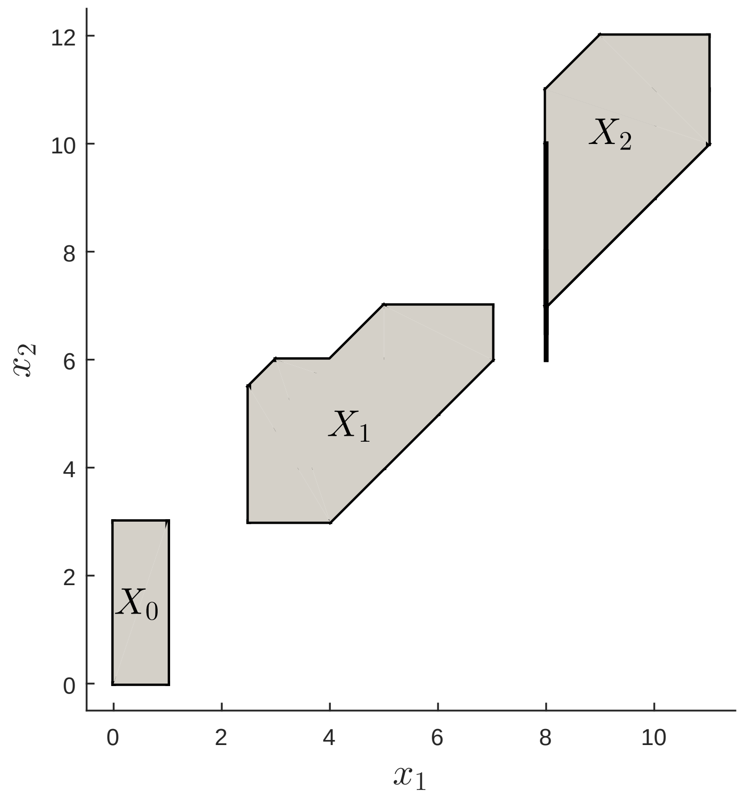

Figure 11 – reach tube for $k \in \{1, 2\}$ (nonautonomous uMPL system).

5.2 Backward Reachability Analysis

Backward reachability analysis of uMPL systems concerns the computation of the set of all states that *may* lead to a given set of final positions via the uMPL dynamics, at a particular event step or over a set of consecutive events. The uMPL systems are defined in an uncertainty context in which the definitions of backward reach sets and backward reach tube are given by:

Definition 5.13 (backward reach set) *Given an uMPL system and a nonempty set of final positions $X_0 \subseteq \mathbb{R}^n$, the **backward reach set** X_{-N} is the set of all states $\boldsymbol{x}(-N)$ that* may *lead to X_0 in N steps of the uMPL dynamics, possibly by application of controls.*

Definition 5.14 (backward reach tube) *Given an uMPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the **reach tube** is defined by the set-valued function $k \mapsto X_{-k}$ for any given $k > 0$ where X_{-k} is defined.*

Remark 5.15 *Note that the definition of backward reach set presented here differs from that* *presented in section 3.2 (see definition 3.7). Consider, for example, that X_{-1} is the backward reach set of a given set of final positions X_0. In the deterministic context (MPL systems), for all $\boldsymbol{x} \in X_{-1}$ we have that $A \otimes \boldsymbol{x} \in X_0$, and therefore $\mathcal{I}_A\{X_{-1}\} \subseteq X_0$. In the uncertain context (uMPL systems), for all $\boldsymbol{x} \in X_{-1}$ it is assured that it is possible to reach X_0 from $\boldsymbol{x}$, i.e., there is at least one $A \in [\boldsymbol{A}]$ such that $A \otimes \boldsymbol{x} \in X_0$. However, in general, this does not hold for all $A \in [\boldsymbol{A}]$, i.e, it may exists some $A \in [\boldsymbol{A}]$ such that $A \otimes \boldsymbol{x} \notin X_0$. Therefore, in the general case, we have that $\mathcal{I}_{[\boldsymbol{A}]}\{X_{-1}\} \nsubseteq X_0$.*

Sections 5.2.1 and 5.2.2 present a procedure to compute the backward reach tube for autonomous and nonautonomous uMPL systems, respectively.

5.2.1 Backward Reachability Analysis of Autonomous uMPL systems

For autonomous uMPL systems, given a set of final positions X_0, the backward reach set X_{-k} at the event step k can be recursively calculated as the inverse image of the reach set X_{-k+1} w.r.t the uMPL dynamics:

$$X_{-k} = \mathcal{I}^{-1}_{[\mathbf{A}]}\{X_{-k+1}\} = \{\mathbf{x} \in \mathbb{R}^n : \exists A \in [\mathbf{A}] : A \otimes \mathbf{x} \in X_{-k+1}\}. \tag{5.20}$$

From Corollary 5.2 it can be shown that if X_0 can be represented by a union of q_0 DBM, then X_{-k} can be represented by a union of q_{-k} DBM, for each $k \in \mathbb{N}$.

Given the set of final conditions X_0, computing the backward reach tube for $k \in \{1, ..., N\}$ can be done as follows: first, construct the partitioned uMPL system generated by $A(k)$; then, for each $k \in \{1, ..., N\}$, compute the inverse image of X_{k-1} w.r.t. the partitioned uMPL system. The worst-case complexity to compute $\mathcal{I}^{-1}_{[\mathbf{A}]}\{X_{-k+1}\}$, for each $k \in \mathbb{N}$ is $\mathcal{O}(q_{-k+1}n^{n+3})$ (see remark 5.3). Thus, the overall complexity is $\mathcal{O}(n^{n+3}\sum_{k=1}^{N} q_{-k+1})$.

The set of all states that may lead to a given set of final positions X_0 in N event steps can be computed using a one-shot procedure. Given a nonempty set of final conditions X_0, the backward reach set X_{-N} is given by:

$$X_{-N} = \mathcal{I}^{-1}_{[\mathbf{A}]^{\otimes N}}\{X_0\} = \{\mathbf{x} \in \mathbb{R}^n : \mathcal{A} \in [\mathbf{A}]^{\otimes N} : \mathcal{A} \otimes \mathbf{x} \in X_0\}. \tag{5.21}$$

A general procedure for computing X_{-N} is: 1) compute $[\mathbf{A}]^{\otimes N}$; then, 2) construct the partitioned uMPL system generated by $[\mathbf{A}]^{\otimes N}$; and, 3) compute the inverse image of X_0 w.r.t. the obtained partitioned system.

Example 5.16 *Consider the autonomous uMPL system given by:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1), \ \text{where } A(k) \in \begin{pmatrix} [4,\ 6] & [3,\ 5] \\ [3,\ 7] & [4,\ 5] \end{pmatrix}.$$

In example 4.11 this system was represented as a collection of DBM $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}\}$.

Given $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \leq x_1 \leq 1,\ 1 \leq x_2 \leq 3\}$, *the backward reach sets* X_{-k} *for* $k \in \{1, 2\}$ *are computed in the following. Note that the set* X_0 *can be represented by the following DBM:*

$$D^{(X_0)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1, \leq) \\ (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

To compute the backward reach set X_{-1} *we must compute the inverse image of* X_0 *w.r.t each component* $\boldsymbol{g} \in \{(1,1),\ (2,1),\ (2,2)\}$ *of the partitioned uMPL system. According to algorithm 5.2, the inverse image of* $D^{(X_0)}$ *w.r.t. a component* $\boldsymbol{g}$ *of the partitioned uMPL system can be computed as follows: first, we compute the Cartesian product of* $D^{(X_0 \times U_1)}$ *and* $D^{(\mathbb{R}^2)}$*:*

$$D^{(X_0 \times \mathbb{R}^2)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array} \end{array}$$

Then we compute the intersection of $D^{(X_0 \times \mathbb{R}^2)}$ *and* $D^{\boldsymbol{g}}$*; next we compute the canonical form of the intersection and finally we project the canonical form over the state variables* x_1 *and* x_2*. For the component* $\boldsymbol{g} = (1, 1)$ *we have that interserction of* $D^{(X_0 \times \mathbb{R}^2)}$ *and* $D^{(1,1)}$ *is given by:*

$$D^{(X_0 \times \mathbb{R}^2)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1, \leq) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (6, \leq) & \varepsilon_{\mathcal{B}} \\ (3, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (7, \leq) & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-4, \leq) & (-3, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-3, \leq) & (-4, \leq) & (1, \leq) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array} \end{array}$$

The canonical form of the intersection is given by:

$$cf(D^{(X_0\times\mathbb{R}^2)}\oplus_{\mathcal{B}} D^{(1,1)}) = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1' & x_2' & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-1,\leq) & (6,\leq) & \varepsilon_{\mathcal{B}} \\ (1,\leq) & e_{\mathcal{B}} & e_{\mathcal{B}} & (6,\leq) & \varepsilon_{\mathcal{B}} \\ (3,\leq) & (3,\leq) & e_{\mathcal{B}} & (7,\leq) & \varepsilon_{\mathcal{B}} \\ (-3,\leq) & (-4,\leq) & (-4,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-2,\leq) & (-3,\leq) & (-4,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \end{array}$$

And the orthogonal projection over the states variables x_1 and x_2 is given by:

$$D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})} = cf(D^{(X_0\times\mathbb{R}^2)}\oplus_{\mathcal{B}} D^{(1,1)})\lceil_{\boldsymbol{x}} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (6,\leq) & \varepsilon_{\mathcal{B}} \\ (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-2,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array}$$

Applying the same procedure to the components $\boldsymbol{g} = (2,1)$ and $\boldsymbol{g} = (2,2)$ we obtain:

$$D^{(X_{-1}|_{\boldsymbol{g}=(2,1)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (6,\leq) & (5,\leq) \\ (-3,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ (-2,\leq) & (2,\leq) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array}$$

$$D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (4,\leq) \\ (-4,\leq) & e_{\mathcal{B}} & (-2,\leq) \\ (-2,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array}$$

Thus, the backward reach set X_{-1} can be represented by the collection of DBM given by $\mathtt{D}^{X_{-1}} = \{D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})},\ D^{(X_{-1}|_{\boldsymbol{g}=(2,1)})},\ D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})}\}$. Moreover, we have that $X_{-1} = \mathcal{R}(D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})}) \cup \mathcal{R}(D^{(X_{-1}|_{\boldsymbol{g}=(2,1)})}) \cup \mathcal{R}(D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})}) = \{\boldsymbol{x}\in\mathbb{R}^2 : -6\leq x_1\leq -3,\ x_2\leq -2,\ x_2-x_1\leq 1\}\cup\{\boldsymbol{x}\in\mathbb{R}^2 : -6\leq x_1\leq -3,\ -5\leq x_2\leq -2,\ 1\leq x_2-x_1\leq 2\}\cup\{\boldsymbol{x}\in\mathbb{R}^2 : x_1\leq -4,\ -4\leq x_2\leq -2,\ x_2-x_1\geq 2\}$.

The backward reach set X_{-2} can be obtained by computing the inverse image of each DBM representing X_{-1} w.r.t each component $\boldsymbol{g}\in\{(1,1),\ (2,1),\ (2,2)\}$ of the partitioned uMPL system, which yields $X_{-2} = \{\boldsymbol{x}\in\mathbb{R}^2 : -12\leq x_1\leq -7,\ x_2\leq -6,\ x_2-x_1\leq 1\}\cup\{\boldsymbol{x}\in\mathbb{R}^2 : -13\leq x_1\leq -7,\ -11\leq x_2\leq -6,\ 1\leq x_2-x_1\leq 2\}\cup\{\boldsymbol{x}\in\mathbb{R}^2 : x_1\leq -8,\ -11\leq x_2\leq -6,\ x_2-x_1\geq 2\}$. The backward reach sets X_{-1} and X_{-2} are shown in Figure 12.

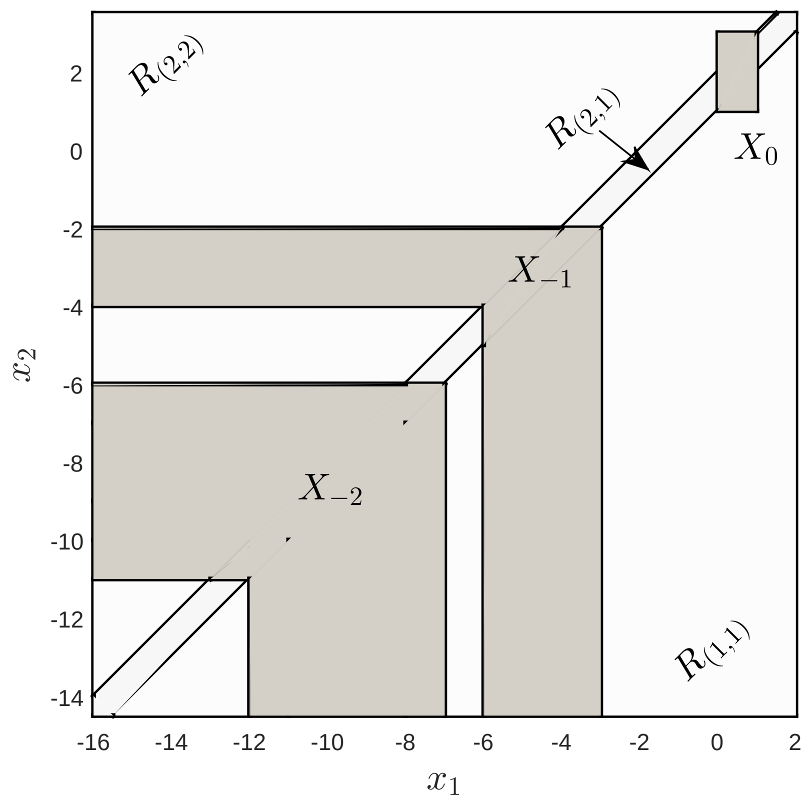

Figure 12 – backward reach tube for $k \in \{1, 2\}$ (autonomous uMPL system).

5.2.2 Backward Reachability Analysis of Nonautonomous uMPL systems

For *non-autonomous* uMPL systems, given a set of final conditions X_0 and the set of inputs U_{-k+1} for each $k \in \mathbb{N}$, the backward reach set X_{-k} at the event step k can be recursively calculated as the inverse image of X_{-k+1}:

$$\begin{aligned} X_{-k} &= \mathcal{I}_F^{-1}\{X_{-k+1}\} \\ &= \{\mathbf{x} \in \mathbb{R}^n : \exists \mathbf{u} \in U_{-k+1},\ \exists F \in [\mathbf{F}] : F \otimes \mathbf{y} \in X_{-k+1}\}. \end{aligned} \tag{5.22}$$

where $[\mathbf{F}] = ([\mathbf{A}]\ \ [\mathbf{B}])$ and $\mathbf{y} = \left(\mathbf{x}^T\ \mathbf{u}^T\right)^T$.

Given X_{-k+1} and U_{-k+1} the backward reach set $X_{-k} = \mathcal{I}_{[\mathbf{F}]}^{-1}\{X_{-k+1}\}$ can be computed as follows: 1) compute de Cartesian product $X_{-k+1} \times \mathbb{R}^n \times U_{-k+1}$; then, 2) intersect the Cartesian product with each component of the partitioned uMPL system generated by $[\mathbf{F}]$; next, 3) compute the canonical form of the intersections, and finally, 4) project the canonical form over the state variables at event step $-k$. The worst-case complexity to compute $\mathcal{I}_{[\mathbf{F}]}^{-1}\{X_{-k+1}\}$ critically depends on the canonical form computation (step 3) and is $\mathcal{O}(\bar{q}_{-k+1}(n+m)^{n+3})$, where: $\bar{q}_{-k+1} = q_{-k+1} r_{-k+1}$; and q_{-k+1} and r_{-k+1} are, respectively, the cardinality of the DBM union set representing X_{-k+1} and U_{-k+1}.

Remark 5.17 *Note that, the Cartesian product of finitely many DBM is a collection of finitely many DBM, the intersection of finitely many DBM is a collection of finitely many DBM, the canonical form of a DBM is a DBM and the projection of a DBM onto a subset of its variables is a DBM. Therefore, if X_{-k+1} and U_{-k+1} can be represented by collections of finitely many DBM then X_{-k} can also be represented by a collection of finitely many DBM. By induction, if X_0 and U_{-k+1}, for each $k \in \mathbb{N}$, can be represented by collections of finitely many DBM, then X_{-k} can also be represented by a collection of finitely many DBM for all $k \in \mathbb{N}$.*

Given X_0 and the set of control inputs U_{-k+1} for each $k \in \mathbb{N}$, the backward reach tube for $k \in \{1, ..., N\}$ can be computed by calculating $X_{-k} = \mathcal{I}^{-1}_{[\mathbf{F}]}\{X_{-k+1}\}$ for $k = 1,\ 2, ..., N$. Thus, the overall complexity to compute backward reach tube is $\mathcal{O}((n+m)^{n+3} \sum_{k=1}^{N} \bar{q}_{-k+1})$.

The following is an extension to uMPL systems of the one-shot procedure for computing the backward reach set X_{-N} presented in section 3.2.2. Given a nonempty set of final conditions X_0, the set of all states that may lead to X_0 in N event steps is given by:

$$\begin{aligned} X_{-N} \quad = \quad & \{\mathbf{x}(-N) \in \mathbb{R}^n : \exists \mathbf{u}(-N+1) \in U_{-N+1}, ..., \mathbf{u}(0) \in U_0 : ([\mathbf{A}]^{\otimes N}, \\ & [\mathbf{A}]^{\otimes(N-1)} \otimes [\mathbf{B}], ..., [\mathbf{B}]) \otimes (\mathbf{x}(-N)^T\ \mathbf{u}(-N+1)^T\ \mathbf{u}(0)^T)^T \in X_0\}. \end{aligned} \tag{5.23}$$

Given the matrices $[\mathbf{A}]$ and $[\mathbf{B}]$, a set of final positions X_0 and a sequence of input sets $U_{-N+1}, ..., U_0$, a general procedure for computing X_{-N} is given by: 1) generate the matrix $([\mathbf{A}]^{\otimes N},\ [\mathbf{A}]^{\otimes(N-1)} \otimes [\mathbf{B}], ..., [\mathbf{B}])$; then, 2) Construct the partitioned uMPL system generated by this matrix; and, 3) compute the inverse image of X_0 w.r.t the obtained partitioned system; 4) intersect the inverse image with $\mathbb{R}^n \times U_1 \times ... \times U_N$; and finally, 5) project the intersection over the state variables. The complexity of this procedure is the same as the one-shot procedure for the forward case presented in section 5.1.2.

Example 5.18 *Consider the nonautonomous uMPL system given by:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1) \oplus B(k) \otimes \boldsymbol{u}(k),$$

where,

$$A(k) \in \begin{pmatrix} 2 & [2,\ 4] \\ [3,\ 5] & [3,\ 4] \end{pmatrix} \text{ and } B(k) \in \begin{pmatrix} e \\ \varepsilon \end{pmatrix}.$$

In example 4.12 this system was represented as a collection of DBM $\mathtt{D} = \{D^{(1,1)}, D^{(2,1)}, D^{(2,2)}, D^{(3,1)}, D^{(3,2)}\}$.

Given $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \le x_1 \le 1,\ 0 \le x_2 \le 3\}$, *and the sequence of controls* $u_1(0) = -0.5$ *and* $u_1(-1) = -7.5$, *the backward reach sets* X_{-k} *for* $k \in \{1, 2\}$ *are computed in the following.*

The set of final positions X_0 *and the control input* $u_1(0)$ *can be represented by the following DBM:*

$$D^{(X_0)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-2, \le) \\ (1, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (5, \le) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \quad D^{(U_0)} = \begin{array}{c} \begin{array}{cc} x_0 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (0.5, \le) \\ (-0.5, \le) & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ u_1 \end{array}$$

In order to compute X_{-1} *we must, first, compute the Cartesian product* $X_0 \times \mathbb{R}^2 \times U_0$, *which can be represented by:*

$$D^{(X_0 \times \mathbb{R}^2 \times U_0)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-2, \le) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (0.5, \le) \\ (1, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (5, \le) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-0.5, \le) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

Then, for each $\boldsymbol{g} \in \{(1, 1),\ (2, 1),\ (2, 2),\ (3, 1),\ (3, 2)\}$, *we compute the intersection of* $D^{(X_0 \times \mathbb{R}^2 \times U_0)}$ *and* $D^{\boldsymbol{g}}$, *the DBM that represents the component* $\boldsymbol{g}$; *next, we compute the canonical form representation of the intersection; and finally, we compute the orthogonal projection of the canonical form over the variables* x_1 *and* x_2.

For the component $\boldsymbol{g} = (1, 1)$ *we have that the intersection of* $D^{(X_0 \times \mathbb{R}^2 \times U_0)}$ *and* $D^{(1,1)}$ *is given by:*

$$D^{(X_0 \times \mathbb{R}^2 \times U_0)} \oplus_{\mathcal{B}} D^{(1,1)} = \begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-2, \le) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (0.5, \le) \\ (1, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2, \le) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (5, \le) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} & (5, \le) & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \le) & (-3, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ \varepsilon_{\mathcal{B}} & (-2, \le) & (-3, \le) & (-2, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-0.5, \le) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2, \le) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array}$$

The canonical form is given by:

$$cf(D^{(X_0\times\mathbb{R}^2\times U_0)} \oplus_{\mathcal{B}} D^{(1,1)}) =$$

$$\begin{array}{c} \begin{array}{cccccc} x_0 & x_1' & x_2' & x_1 & x_2 & u_1 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & (-2,\leq) & (2,\leq) & \varepsilon_{\mathcal{B}} & (0.5,\leq) \\ (1,\leq) & e_{\mathcal{B}} & (-1,\leq) & (2,\leq) & \varepsilon_{\mathcal{B}} & (1.5,\leq) \\ (4,\leq) & (3,\leq) & e_{\mathcal{B}} & (5,\leq) & \varepsilon_{\mathcal{B}} & (4.5,\leq) \\ (-1,\leq) & (-2,\leq) & (-3,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-0.5,\leq) \\ (-3,\leq) & (-4,\leq) & (-5,\leq) & (-2,\leq) & e_{\mathcal{B}} & (-2.5,\leq) \\ (-0.5,\leq) & (-0.5,\leq) & (-2.5,\leq) & (1.5,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1' \\ x_2' \\ x_1 \\ x_2 \\ u_1 \end{array} \end{array}$$

And the orthogonal projection of the canonical form over the state variables x_1 and x_2 is given by:

$$D^{(X_{-1}|_{\boldsymbol{g}=(1,1)})} = cf(D^{(X_0\times\mathbb{R}^2\times U_0)} \oplus_{\mathcal{B}} D^{(1,1)})\lceil_{\boldsymbol{x}} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (2,\leq) & \varepsilon_{\mathcal{B}} \\ (-1,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-3,\leq) & (-2,\leq) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

Applying the same procedure to the components $\boldsymbol{g} = (2,1)$, $\boldsymbol{g} = (2,2)$, $\boldsymbol{g} = (3,1)$ and $\boldsymbol{g} = (3,2)$ we obtain:

$$D^{(X_{-1}|_{\boldsymbol{g}=(2,1)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (3,\leq) & (4,\leq) \\ (-1,\leq) & e_{\mathcal{B}} & (2,\leq) \\ (-1,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

$$D^{(X_{-1}|_{\boldsymbol{g}=(2,2)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (2,\leq) \\ (-2,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ (-1,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

$$D^{(X_{-1}|_{\boldsymbol{g}=(3,1)})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} \top_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-1,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-1,\leq) & (1,\leq) & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

$$D^{(X_{-1}|_{g=(3,2)})} = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{ccc} x_0 & x_1 & x_2 \\ \end{array}$$

$$D^{(X_{-1}|_{g=(3,2)})} = \begin{array}{cc} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ & \begin{pmatrix} \top_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (-1,\leq) & e_{\mathcal{B}} & (-1,\leq) \\ (-1,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array} \end{array}$$

Note that $D^{(X_{-1}|_{g=(3,1)})}$ and $D^{(X_{-1}|_{g=(3,2)})}$ are empty DBM, and therefore the inverse image of X_0 w.r.t. the components $\boldsymbol{g} = (3,1)$ and $\boldsymbol{g} = (3,2)$ is empty. Thus, the backward reach set X_{-1} can be represented by the collection of DBM given by $\mathtt{D}^{X_{-1}} = \{D^{(X_{-1}|_{g=(1,1)})}$, $D^{(X_{-1}|_{g=(2,1)})}$, $D^{(X_{-1}|_{g=(2,2)})}\}$. Moreover, we have that $X_{-1} = \mathcal{R}(D^{(X_{-1}|_{g=(1,1)})}) \cup \mathcal{R}(D^{(X_{-1}|_{g=(2,1)})}) \cup \mathcal{R}(D^{(X_{-1}|_{g=(2,2)})}) = \{\boldsymbol{x} \in \mathbb{R}^2 : -2 \leq x_1 \leq -1,\ x_2 \leq -3,\ x_2 - x_1 \leq -2\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : -3 \leq x_1 \leq -1,\ -4 \leq x_2 \leq -1,\ -2 \leq x_2 - x_1 \leq 1\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \leq -2,\ -2 \leq x_2 \leq -1,\ x_2 - x_1 \geq 1\}$.

The backward reach set X_{-2} can be obtained by computing the inverse image of each DBM representing X_{-1} w.r.t each component $\boldsymbol{g} \in \{(1,1), (2,1), (2,2), (3,1), (3,2)\}$ of the partitioned uMPL system, which yields $X_{-2} = \{\boldsymbol{x} \in \mathbb{R}^2 : -7 \leq x_1 \leq -4,\ x_2 \leq -4,\ x_2 - x_1 \leq 0\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \leq -4,\ -7 \leq x_2 \leq -4,\ x_2 - x_1 \geq 0\}$. The backward reach sets X_{-1} and X_{-2} are shown in Figure 13.

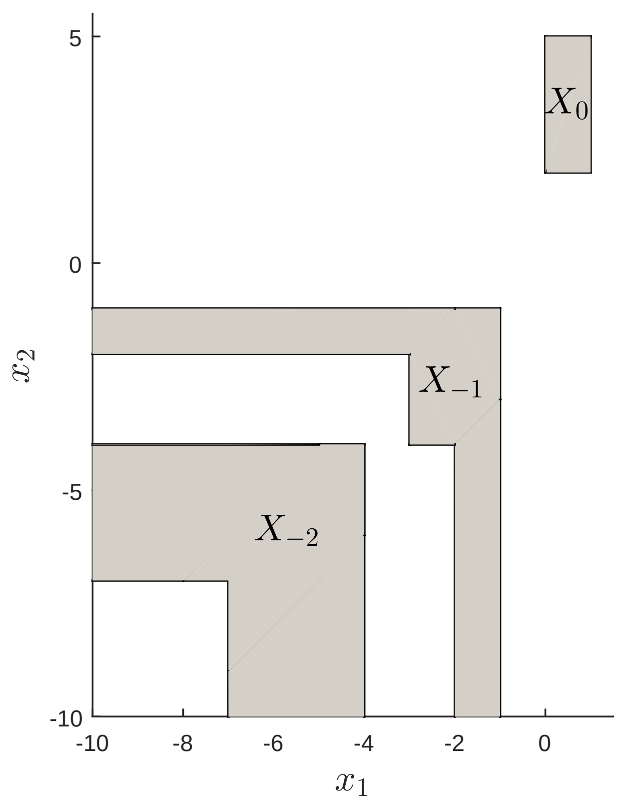

Figure 13 – backward reach tube for $k \in \{1, 2\}$ (nonautonomous uMPL system).

5.3 Image and inverse image of a Point

Every point in $\mathbb{R}^n$ can be represented by a DBM in $\mathcal{B}^{(n+)\times(n+1)}$, and therefore the procedures presented in sections 5.1 and 5.2 can be used to compute the image and the inverse image of a point w.r.t. an uMPL system. However computing the image and the inverse image of a point w.r.t. an uMPL system can be done by considering a less expensive approach.

In the following sections, we present alternative procedures to compute the image and the inverse image of a point w.r.t. a generic uMPL system given by:

$$\mathbf{z}(k) = A \otimes \mathbf{x}(k-1),\ A \in [\mathbf{A}],\ A \in \mathbb{R}^{n\times p} \tag{5.24}$$

5.3.1 Image of a Point

In section 4.1 it was demonstrated that given $\mathbf{x}(k-1)$ then $z_i(k)$ is in the interval defined by:

$$[\mathbf{z}_i](k) = \left[\bigoplus_{j=1}^{p}\{\underline{a}_{ij} \otimes x_j(k-1)\},\ \bigoplus_{j=1}^{p}\{\overline{a}_{ij} \otimes x_j(k-1)\}\right].$$

Therefore, it is straightforward to see that the image of a point $\mathbf{x}$ w.r.t the uMPL system is given by:

$$\mathcal{I}_{[\mathbf{A}]}\{\mathbf{x}\} = \bigcap_{i=1}^{n}\left\{\mathbf{z} \in \mathbb{R}^n : \bigoplus_{j=1}^{p}\underline{a}_{ij} \otimes x_j \le z_i \le \bigoplus_{j=1}^{p}\overline{a}_{ij} \otimes x_j\right\} \tag{5.25}$$

Or equivalently,

$$\mathcal{I}_{[\mathbf{A}]}\{\mathbf{x}\} = \left\{\mathbf{z} \in \mathbb{R}^n : \underline{A} \otimes \mathbf{x} \le \mathbf{z} \le \overline{A} \otimes \mathbf{x}\right\} \tag{5.26}$$

Remark 5.19 *Note that the image of a point w.r.t an uMPL system is a hyperrectangle. Although this kind of set can be represented and manipulated using a simpler data structure, we will keep the DBM. This can be useful if we have a set of initial positions X_0 given by a single point and we want to compute a reach set for some $k > 1$. In this case we could compute the reach set X_1 using equation (5.25) and the next reach sets would be calculated using the procedure presented in section 5.1.*

Example 5.20 *Consider the following uMPL system:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1), \quad A(k) \in [\boldsymbol{A}],$$

where

$$[\boldsymbol{A}] = \begin{pmatrix} [1,\ 3] & 2 \\ [2,\ 4] & [3,\ 6] \end{pmatrix}.$$

Given $\boldsymbol{x}(0) = (0\ 0)^T$*, we have that,*

$$\begin{aligned} \mathcal{I}_{[\boldsymbol{A}]}\{\boldsymbol{x}(0)\} &= \Big\{\boldsymbol{x} \in \mathbb{R}^2 : (1 \otimes 0) \oplus (2 \otimes 0) \leq x_1 \leq (3 \otimes 0) \oplus (2 \otimes 0)\Big\} \\ &\qquad \cap \Big\{\boldsymbol{x} \in \mathbb{R}^2 : (2 \otimes 0) \oplus (3 \otimes 0) \leq x_2 \leq (4 \otimes 0) \oplus (6 \otimes 0)\Big\} \\ &= \Big\{\boldsymbol{x} \in \mathbb{R}^2 : 2 \leq x_1 \leq 3\Big\} \cap \Big\{\boldsymbol{x} \in \mathbb{R}^2 : 3 \leq x_2 \leq 6\Big\} \end{aligned}$$

Moreover, this set can be represented by the following DBM:

$$D^{(X_1)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-2, \leq) & (-3, \leq) \\ (3, \leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (6, \leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} x_0 \\ x_1 \\ x_2 \end{array}$$

5.3.2 Inverse Image of a Point

The inverse image of a given point $\mathbf{z}$ w.r.t. an uMPL system is defined as:

$$\mathcal{I}^{-1}_{[\mathbf{A}]}\{\mathbf{z}\} = \{\mathbf{x} \in \mathbb{R}^p : \exists A \in [\mathbf{A}] : A \otimes \mathbf{x} = \mathbf{z}\}. \tag{5.27}$$

Equivalently, it can be stated that $\mathbf{x} \in \mathcal{I}^{-1}_{[\mathbf{A}]}\{\mathbf{z}\}$ if and only if $\mathbf{z}$ is in the image of $\mathbf{x}$ w.r.t the uMPL system , i.e.,

$$\mathbf{x} \in \mathcal{I}^{-1}_{[\mathbf{A}]}\{\mathbf{z}\} \Leftrightarrow \underline{A} \otimes \mathbf{x} \leq \mathbf{z} \leq \overline{A} \otimes \mathbf{x}. \tag{5.28}$$

Thus, $\mathbf{x}$ has to satisfy two restictions:

$$\underline{A} \otimes \mathbf{x} \ \leq \ \mathbf{z}, \tag{5.29}$$

$$\overline{A} \otimes \mathbf{x} \ \geq \ \mathbf{z}. \tag{5.30}$$

Then, the inverse image of a point $\mathbf{z}$ can be represented by the intersection of two sets:

$$\mathcal{I}^{-1}_{[\mathbf{A}]}\{\mathbf{z}\} = U \cap L \tag{5.31}$$

where U is the set of all $\mathbf{x}$ that satisfies (5.29) and L is the set of all $\mathbf{x}$ that satisfies (5.30).

By using residuation (see section 2.2), it can be demonstrated that the set U is given by:

$$U = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{x} \leq \underline{A} \diamond \mathbf{z}\}. \tag{5.32}$$

where $\diamond\!\!\!\backslash$ is the residuation operator.

On the other hand, the set L can be expressed as:

$$L = \bigcap_{i=1}^{n} \left\{ \mathbf{x} \in \mathbb{R}^p : z_i \leq \overline{A}[i,:] \otimes \mathbf{x} \right\}, \tag{5.33}$$

where $\overline{A}[i,:]$ is the i-th row of matrix $\overline{A}$.

We seek for a representation of L in which $\mathbf{x}$ is not implicit. In this sense, we compute the complement of L, which is given by:

$$L^c = \bigcup_{i=1}^{n} \left\{ \mathbf{x} \in \mathbb{R}^p : \overline{A}[i,:] \otimes \mathbf{x} < z_i \right\} \tag{5.34}$$

By using residuation we have that:

$$L^c = \bigcup_{i=1}^{n} \left\{ \mathbf{x} \in \mathbb{R}^p : \mathbf{x} < \underline{X}^{(i)} \right\} \tag{5.35}$$

where,

$$\underline{X}^{(i)} = \overline{A}[i,:] \diamond\!\!\!\backslash z_i. \tag{5.36}$$

Equivalently, equation (5.35) can be expressed as:

$$L^c = \bigcup_{i=1}^{n} \left(\bigcap_{j=1}^{p} \left\{ \mathbf{x} \in \mathbb{R}^p : x_j < \underline{X}_j^{(i)} \right\} \right). \tag{5.37}$$

The set L can be obtained by computing the complement of L^c, i.e., $L = (L^c)^c$. The complement of L^c is computed in the following.

Consider the intersection of n sets noted by $\bigcap_{j=1}^{n} \mathcal{A}_j$. The complement of the intersection is given by:

$$\left(\bigcap_{j=1}^{n} \mathcal{A}_j \right)^c = \bigcup_{j=1}^{n} \mathcal{A}_j^c. \tag{5.38}$$

However, if we want to represent the complement by a union of pairwise disjoint sets, equation (5.38) can be expressed as:

$$\begin{aligned} \left(\bigcap_{j=1}^{n} \mathcal{A}_j \right)^c &= \mathcal{A}_1^c \cup [\mathcal{A}_1 \cap \mathcal{A}_2^c] \cup [\mathcal{A}_1 \cap \mathcal{A}_2 \cap \mathcal{A}_3^c] \cup \cdots \cup \left[\left(\bigcap_{k=1}^{n-1} \mathcal{A}_k \right) \cap \mathcal{A}_n^c \right] \\ &= \bigcup_{j=1}^{n} \left[\left(\bigcap_{k=1}^{j-1} \mathcal{A}_k \right) \cap \mathcal{A}_j^c \right]. \end{aligned} \tag{5.39}$$

where $\bigcap_{k=1}^{0} \mathcal{A}_k$ is set to $\mathbb{R}^p$.

Example 5.21 *Let us compute the complement of the set* $\{\boldsymbol{x} \in \mathbb{R}^3 : \boldsymbol{x} < \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T\}$. *This set can be expressed as* $\underbrace{\{\boldsymbol{x} \in \mathbb{R}^3 : x_1 < 0\}}_{\mathcal{A}_1} \cap \underbrace{\{\boldsymbol{x} \in \mathbb{R}^3 : x_2 < 0\}}_{\mathcal{A}_2} \cap \underbrace{\{\boldsymbol{x} \in \mathbb{R}^3 : x_3 < 0\}}_{\mathcal{A}_3}$. *Thus, according to* (5.39), *we have that:*

$$\begin{aligned}\{\boldsymbol{x} \in \mathbb{R}^3 : \boldsymbol{x} < \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T\}^c &= \{\boldsymbol{x} \in \mathbb{R}^3 : x_1 \geq 0\} \\ &\cup \Big[\{\boldsymbol{x} \in \mathbb{R}^3 : x_1 < 0\} \cap \{\boldsymbol{x} \in \mathbb{R}^3 : x_2 \geq 0\}\Big] \\ &\cup \Big[\{\boldsymbol{x} \in \mathbb{R}^3 : x_1 < 0\} \cap \{\boldsymbol{x} \in \mathbb{R}^3 : x_2 < 0\} \cap \{\boldsymbol{x} \in \mathbb{R}^3 : x_3 \geq 0\}\Big]\end{aligned}$$

Therefore, according to (5.39), the complement each term of union (5.37) can be computed as:

$$\left(\bigcap_{j=1}^{p} \underbrace{\left\{\mathbf{x} \in \mathbb{R}^p : x_j < \underline{X}_j^{(i)}\right\}}_{\mathcal{A}_j}\right)^c = \bigcup_{j=1}^{p} \left[\left(\bigcap_{k=1}^{j-1} \underbrace{\left\{\mathbf{x} \in \mathbb{R}^p : x_k < \underline{X}_k^{(i)}\right\}}_{\mathcal{A}_k}\right) \cap \underbrace{\left\{\mathbf{x} \in \mathbb{R}^p : x_j \geq \underline{X}_j^{(i)}\right\}}_{\mathcal{A}_j^c}\right] \tag{5.40}$$

where $\bigcap_{k=1}^{0} \left\{\mathbf{x} \in \mathbb{R}^p : x_k < \underline{X}_k^{(i)}\right\}$ is set to $\mathbb{R}^p$.

Then, the complement of L^c is:

$$\begin{aligned} L = (L^c)^c &= \left[\bigcup_{i=1}^{n} \left(\bigcap_{j=1}^{p} \left\{\mathbf{x} \in \mathbb{R}^p : x_j < \underline{X}_j^{(i)}\right\}\right)\right]^c \\ &= \bigcap_{i=1}^{n} \left[\left(\bigcap_{j=1}^{p} \left\{\mathbf{x} \in \mathbb{R}^p : x_j < \underline{X}_j^{(i)}\right\}\right)^c\right] \\ &= \bigcap_{i=1}^{n} \left(\bigcup_{j=1}^{p} \left[\left(\bigcap_{k=1}^{j-1} \left\{\mathbf{x} \in \mathbb{R}^p : x_k < \underline{X}_k^{(i)}\right\}\right) \cap \left\{\mathbf{x} \in \mathbb{R}^p : x_j \geq \underline{X}_j^{(i)}\right\}\right]\right) \end{aligned} \tag{5.41}$$

Defining:

$$set_j^i = \bigcap_{k=1}^{j-1} \left\{\mathbf{x} \in \mathbb{R}^p : x_k < \underline{X}_k^{(i)}\right\} \cap \left\{\mathbf{x} \in \mathbb{R}^p : x_j \geq \underline{X}_j^{(i)}\right\}, \tag{5.42}$$

we have that:

$$\begin{aligned} L &= \bigcap_{i=1}^{n} \bigcup_{j=1}^{p} set_j^i \\ &= \left(set_1^1 \cup \cdots \cup set_p^1\right) \cap \left(set_1^2 \cup \cdots \cup set_p^2\right) \cap \cdots \cap \left(set_1^m \cup \cdots \cup set_p^n\right) \\ &= \left(set_1^1 \cap set_1^2 \cap \cdots \cap set_1^n\right) \cup \left(set_1^1 \cap set_1^2 \cap \cdots \cap set_2^n\right) \cup \cdots \cup \left(set_p^1 \cap set_p^2 \cap \cdots \cap set_p^n\right) \end{aligned} \tag{5.43}$$

Now, let us define:

$$SET^{\mathbf{g}} = \bigcap_{i=1}^{n} set^{i}_{g_i}, \ g_i \in \{1, ..., p\} \tag{5.44}$$

Thus, from (5.43), the region L can be expressed as:

$$L = \bigcup_{\mathbf{g} \in \{1,...,p\}^n} SET^{\mathbf{g}} \tag{5.45}$$

Then, from (5.31), we have that:

$$\begin{aligned} \mathcal{I}^{-1}_{[\mathbf{A}]}\{\mathbf{z}\} &= U \cap \bigcup_{\mathbf{g} \in \{1,...,p\}^n} SET^{\mathbf{g}} \\ &= \bigcup_{\mathbf{g} \in \{1,...,p\}^n} \left(SET^{\mathbf{g}} \cap U\right), \end{aligned} \tag{5.46}$$

where U is defined by (5.32).

Note that, the inverse image of a point w.r.t. an uMPL system can be represented by a collection of pairwise disjoint hyperrectangles.

Example 5.22 *Consider the autonomous uMPL system given by:*

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1),$$

where,

$$A(k) \in \begin{pmatrix} [1, \ 4] & [2, \ 3] \\ [1, \ 2] & [0, \ 4] \end{pmatrix}.$$

Given $\boldsymbol{x}(1) = (5, \ 4)^T$*, let us compute* $X_0 = \mathcal{I}^{-1}_{[\boldsymbol{A}]}\{\boldsymbol{x}(1)\} = \bigcup_{\boldsymbol{g} \in \{1,2\}^2} (SET^{\boldsymbol{g}} \cap U)$*. According to equation* (5.32)*, the set* U *is given by:* $U = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \leq \underline{A} \diamond \boldsymbol{x}(1)\}$ *, where,*

$$\underline{A} \diamond \boldsymbol{x}(1) \ = \ \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix} \diamond \begin{pmatrix} 5 \\ 4 \end{pmatrix} = \begin{pmatrix} (5-1) \wedge (4-1) \\ (5-2) \wedge (4-0) \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}.$$

Thus, $U = \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \leq 3, \ x_2 \leq 3\}$*. In order to compute the sets* set^i_j*,* $i, j \in \{1, 2\}$ *, we must compute first* $\underline{X}^{(i)} = \overline{A}[i, :] \diamond x_i(1)$*, for* $i \in \{1, 2\}$*:*

$$\begin{aligned} \underline{X}^{(1)} &= \overline{A}[1, :] \diamond x_1(1) = \begin{pmatrix} 4 & 3 \end{pmatrix} \diamond \begin{pmatrix} 5 \end{pmatrix} = \begin{pmatrix} (5-4) \\ (5-3) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ \underline{X}^{(2)} &= \overline{A}[2, :] \diamond x_2(1) = \begin{pmatrix} 2 & 4 \end{pmatrix} \diamond \begin{pmatrix} 4 \end{pmatrix} = \begin{pmatrix} (4-2) \\ (4-4) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}. \end{aligned}$$

According to (5.42)*, the sets* set_j^i*,* $i, j \in \{1, 2\}$*, are given by:*

$$\begin{array}{ll} set_1^1 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \geq 1\right\} & set_2^1 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 < 1,\ x_2 \geq 2\right\} \\ set_1^2 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \geq 2\right\} & set_2^2 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 < 2,\ x_2 \geq 0\right\} \end{array}$$

Now, for each $\boldsymbol{g} \in \{1, 2\}^2 = \{(1, 1),\ (1, 2),\ (2, 1),\ (2, 2)\}$ *we compute the sets* $SET^{\boldsymbol{g}}$ *as follows:*

$$\begin{array}{rcl} SET^{(1,1)} & = & set_1^1 \cap set_1^2 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 \geq 2\right\}, \\ SET^{(1,2)} & = & set_1^1 \cap set_2^2 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : 1 \leq x_1 < 2,\ x_2 \geq 0\right\}, \\ SET^{(2,1)} & = & set_2^1 \cap set_1^2 = \emptyset, \\ SET^{(2,2)} & = & set_2^1 \cap set_2^2 = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 < 1,\ x_2 \geq 2\right\}. \end{array}$$

Finally we compute $X_0 = \bigcup_{\boldsymbol{g} \in \{1,\ldots,p\}^n} (SET^{\boldsymbol{g}} \cap U)$ *(see* (5.46)*):*

$$\begin{array}{rcl} X_0 & = & (SET^{(1,1)} \cap U) \cup (SET^{(1,2)} \cap U) \cup (SET^{(2,2)} \cap U) \\ & = & \left\{\boldsymbol{x} \in \mathbb{R}^2 : 2 \leq x_1 \leq 3, x_2 \leq 3\right\} \cup \left\{\boldsymbol{x} \in \mathbb{R}^2 : 1 \leq x_1 < 2,\ 0 \leq x_2 \leq 3\right\} \\ & & \cup \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 < 1,\ 2 \leq x_2 \leq 3\right\} \end{array}$$

The invserse image of $\boldsymbol{x}(1)$ *can be observed in Figure 14. Note that* X_0 *is a union of pairwise disjoint hyperrectangles.*

Remark 5.23 *Back to the discussion presented in remark 5.19, we will keep the DBM data structure to represent the hyperrectangles. If a DBM* $D \in \mathcal{B}^{n \times n}$ *represent a hyperrectangle then all non-redundant constraints are in its first row/column. In this case, the checking for emptiness can be performed by verifying if exists an* $i \in \{1, ..., n\}$ *such that*[4] $d_{1i} \otimes_{\mathcal{B}} d_{i1} \succ_{\mathcal{B}} e_{\mathcal{B}}$*. If so, there will be a constraint* $\underline{x}_i \leq x_i \leq \overline{x}_i$ *such that* $\underline{x}_i > \overline{x}_i$ *and therefore the DBM represents an empty region. Note that if a DBM represents a hyperrectangle the checking for emptiness does not require the computation of the canonical form representation, therefore the complexity reduces from cubic to linear w.r.t. its dimension. Furthermore, the intersection of two DBM representing a hyperrectangle can be done with linear complexity w.r.t. its dimension, instead of the quadratic complexity for general DBM.*

Algorithm 5.3 describes a general procedure for computing the inverse image of a point w.r.t an uMPL system using the DBM data structure. The worst-case complexity

[4] The order $\succ$ in $\mathcal{B}$ coincides with the usual lexicographic order $<$ (see remark 2.17)

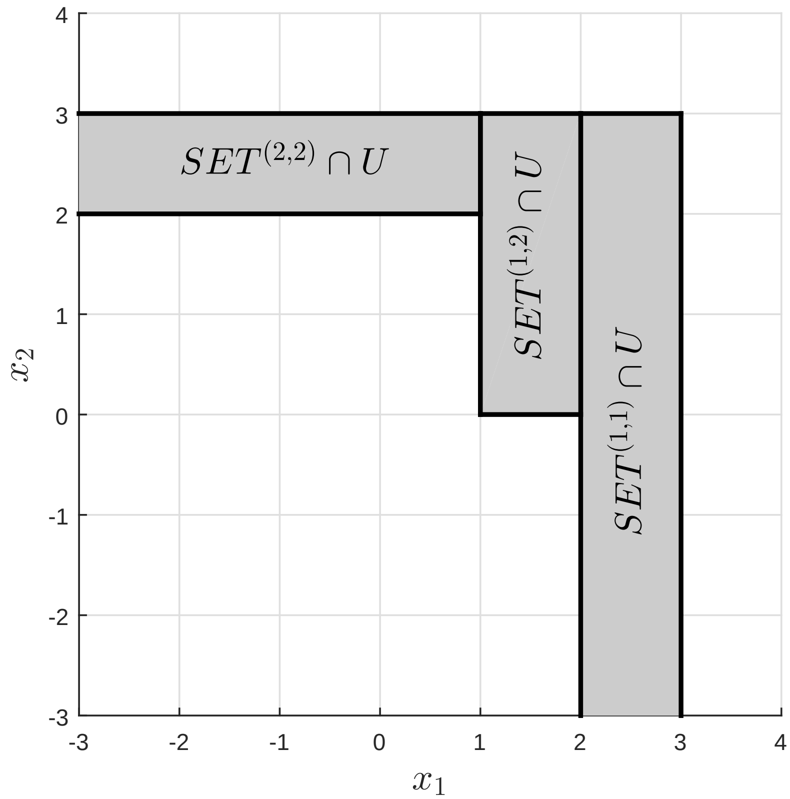

Figure 14 – Inverse image of $\mathbf{x}(1)$.

of the Algorithm critically depends on step 18 and is calculated as follows: the worst-case complexity of step 19 is $\mathcal{O}(p^n)$, the complexity of steps 21 and 22 amounts to $\mathcal{O}(np)$ and the complexity of step 25 is $\mathcal{O}(p)$ (see remark 5.23). Therefore, the worst-case complexity of the Algorithm is $\mathcal{O}(np^{n+1})$.

Remark 5.24 *For autonomous uMPL systems, parameter p equals n, and therefore the worst-case complexity of Algorithm 5.3 is $\mathcal{O}(n^{n+2})$. For nonautonomous uMPL systems, parameter p equals $n + m$, and therefore the worst-case complexity is $\mathcal{O}(n(n + m)^{n+1})$. Note that the worst case complexity of computing the inverse image of a DBM w.r.t a partitioned uMPL system generated by an uMPL system is $\mathcal{O}(n^{n+3})$ for autonomous uMPL systems and $\mathcal{O}((n + m)^{n+3})$ for nonautonomous uMPL systems (see Remark 5.3).*

Algorithm 5.3: Inverse image of a point w.r.t an uMPL system

input : $\mathbf{z} \in \overline{\mathbb{R}}^{n}_{max}$, $\underline{A} \in \overline{\mathbb{R}}^{n\times p}_{max}$, $\overline{A} \in \overline{\mathbb{R}}^{n\times p}_{max}$
output: `D` // A collection of DBM representing $\mathcal{I}^{-1}_{[\mathtt{A}]}\{\mathbf{z}\}$;

begin // Compute the set U and represent it as the DBM $D^{(U)}$;
  $\overline{X} \leftarrow \underline{A} \backslash\!\!\!\circ \{\mathbf{z}\}$ // see (2.27)
  $D^{(U)} \leftarrow e^{(p+1)\times(p+1)}$;
  for $j \in \{1, ..., p\}$ **do** $D^{(U)}[j+1, 1] \leftarrow (\overline{X}_j, \leq)$;
end
begin // Compute the sets set^i_j and represent them as DBM $D^{(set^i_j)}$;
  for all $i \in \{1, ..., n\}$ **do**
    $\underline{X}^{(i)} \leftarrow \overline{A}[i, :] \backslash\!\!\!\circ \{\mathbf{z}\}$ // see (2.27)
    for all $j \in \{1, ..., p\}$ **do**
      $D^{(set^i_j)} \leftarrow e^{(p+1)\times(p+1)}$;
      $D^{(set^i_j)}[1, j+1] \leftarrow (-\underline{X}^{(i)}_j, \leq)$; // represents $\left\{\mathbf{x} \in \mathbb{R}^p : x_j \geq \underline{X}^{(i)}_j\right\}$;
      for all $k \in \{1, ..., (j-1)\}$ **do** // represent $\bigcap_{k=1}^{j-1} \left\{\mathbf{x} \in \mathbb{R}^p : x_k < \underline{X}^{(i)}_k\right\}$;
        $D^{(set^i_j)}[k+1, 1] \leftarrow (\underline{X}^{(i)}_k, <)$;
      end for
    end for
  end for
end
begin // Compute the DBM union set `D` representing $\bigcup_{\mathbf{g}\in\{1,...,p\}^n} (SET^{\mathbf{g}} \cap U)$;
  for all $g \in \{1, ..., p\}^n$ **do**
    $D^{(SET^{\mathbf{g}})} \leftarrow e^{(p+1)\times(p+1)}$;
    for all $i \in \{1, ..., n\}$ **do** // represent $SET^{\mathbf{g}} = \bigcap_{i=1}^{n} set^i_{g_i}$
      $D^{(SET^{\mathbf{g}})} \leftarrow D^{(SET^{\mathbf{g}})} \oplus_{\mathcal{B}} D^{(set^i_{g_i})}$;
    end for
    $D^{(SET^{\mathbf{g}} \cap U)} \leftarrow D^{(SET^{\mathbf{g}})} \oplus_{\mathcal{B}} D^{(U)}$// represent $SET^{\mathbf{g}} \cap U$, see (5.46);
    if $D^{(SET^{g} \cap U)}$ *is not empty* **then**
      $\mathtt{D} \leftarrow \mathtt{D} \cup \{D^{(SET^{\mathbf{g}} \cap U)}\}$;
    end if
  end for
end

6 Application: Conditional Reachability Analysis of uMPL Systems

This chapter presents an application of reachability analysis of uMPL systems. We define the conditional reachability problem and then we show that this problem can be solved by using the results presented in chapter 5.

6.1 The Conditional Reachability Problem

Bayesian methods provide a rigorous general framework for dynamic state estimation problems (GORDON *et al.*, 1993). Consider the following system:

$$\mathbf{x}(k) = f_{k-1}\left(\mathbf{x}(k-1), \mathbf{w}(k)\right), \tag{6.1}$$

$$\mathbf{z}(k) = h_k\left(\mathbf{x}(k), \mathbf{v}(k)\right). \tag{6.2}$$

Where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^l$ are, respectively, the state and measurement vectors; $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^r$ are independent identically distributed (iid) process noise sequence; $f_{k-1} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is, in general, a nonlinear transition function and $h_k : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^l$ is the measurement function.

In the Bayesian approach, one aims to construct the posterior Probability Density Function (PDF) $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$, which is the PDF of the states $\mathbf{x}(k)$ given all the available information $\mathbf{z}(1), ..., \mathbf{z}(k)$ at the event step k. The posterior PDF may be obtained recursively in two stages: prediction and update (GORDON *et al.*, 1993). In the prediction stage it is assumed that the required PDF $p(\mathbf{x}_{k-1}|\mathbf{z}_1, ..., \mathbf{z}_{k-1})$ is available at the event step $k-1$. Using the system model and the *Chapman-Kolmogorov* equation it is possible to obtain the *prior* PDF $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_{k-1})$ based on all information available at the event step $k-1$. In the update stage, the required PDF $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$ is obtained by updating the prior PDF, via the Bayes rule, based on the new available information $\mathbf{z}_k$ and on the measurement model.

In this work, the system described by equations (6.1) and (6.2) are assumed to be an uMPL system, i.e:

$$\mathbf{x}(k) = A(k) \otimes \mathbf{x}(k-1),\ A(k) \in [\mathbf{A}], \tag{6.3}$$

$$\mathbf{z}(k) = C(k) \otimes \mathbf{x}(k),\ C(k) \in [\mathbf{C}]. \tag{6.4}$$

The elements of matrices $A(k) \in \mathbb{R}^{n\times n}$ and $C(k) \in \mathbb{R}^{l\times n}$ are associated to stochastic processes with supports in real intervals. No further assumptions are made on these processes.

The calculation of the support of $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$ is closely related to the *conditional reachability problem*. The conditional reachability analysis concerns the computation of the set of all states that may be reached from a set of initial states, conditioned to a sequence of measures. This set will be called the conditional reach set and is formally defined as follows:

Definition 6.1 (conditional reach sets) *Given a set of initial positions X_0 and a sequence of measures $\{\boldsymbol{z}(1), ..., \boldsymbol{z}(N)\}$, the* conditional reach set *$X_{N|N}$, at event step N, is the set of all states that may be reached from $X_{N-1|N-1}$ (the conditional reach set at $N-1$) via the uMPL transition model* (6.3) *and that may lead to $\boldsymbol{z}(N)$ via the uMPL measurement model in one event step*(6.4)*.*

Note that the conditional reach set $X_{k|k}$, at event step k, corresponds to the exact support of $p(\mathbf{x}_k|\mathbf{z}_1, ..., \mathbf{z}_k)$. Moreover, note that the conditional reachability problem is not stochastic since it does not lead to an estimate (in the estimation theory sense) of any probabilistic parameter. Although not stochastic, the conditional reachability analysis could come in handy, for instance, in the improvement of particle filtering algorithms. Particle Filters, or Sequential Monte Carlo methods, are suboptimal Bayesian algorithms based on weighted-particles approximation of probability densities (ARULAMPALAM *et al.*, 2002; DOUCET *et al.*, 2000). Particle filters applied to Max Plus systems have been studied in (SILVA *et al.*, 2011; CÂNDIDO *et al.*, 2013; CÂNDIDO; MENDES, 2014). In the particle filtering process is common to obtain a set of weighted-particles representing an approximation for a PDF, in which several particles have null weight. These particles does not contribute to the approximation of the PDF. Indeed, particles with null weight are characterized to be outside the support of the PDF. In this context, conditional reachability analysis could be used in the development of procedures to obtain particles inside the support of the PDF, which improves the approximation quality.

As will be shown in the following section it is possible to compute the conditional reach sets by using reachability analysis of uMPL systems.

6.2 The Solution

Assuming that conditional reach set $X_{k-1|k-1}$ is known at the event step $k-1$, and given the measurement $\mathbf{z}(k)$, the conditional reach set $X_{k|k}$ can be calculated in two stages: In the first stage it is computed the image of $X_{k-1|k-1}$ w.r.t. the uMPL transition model, which can be calculated via (5.11) for autonomous uMPL sytems:

$$X_{k|k-1} = \mathcal{I}_{[\mathbf{A}]}\{X_{k-1|k-1}\} = \{A \otimes \mathbf{x} : \mathbf{x} \in X_{k-1|k-1},\, A \in [\mathbf{A}]\}, \tag{6.5}$$

and via (5.18) for *nonautonomous* uMPL systems:

$$\begin{aligned} X_{k|k-1} &= \mathcal{I}_{[\mathbf{F}]}\{X_{k-1|k-1} \times U_k\} \\ &= \{F \otimes \mathbf{y} : \mathbf{y} \in X_{k-1|k-1} \times U_k,\ F \in [\mathbf{F}]\}. \end{aligned} \tag{6.6}$$

Remark 6.2 *Note that the set $X_{k|k-1}$ corresponds to the support of the prior PDF $p(\boldsymbol{x}_k|\boldsymbol{z}_1, ..., \boldsymbol{z}_{k-1})$. In this sense, the first stage can be associated to the prediction stage of the Bayesian approach.*

The second stage is subdivided in two sub-stages: In the first sub-stage, it is computed the inverse image of $\mathbf{z}(k)$ w.r.t. the uMPL measurement model, which can be calculated via (5.27) :

$$\tilde{X}_{k|k} = \mathcal{I}^{-1}_{[\mathbf{C}]}\{\mathbf{z}_k\} = \{\mathbf{x} \in \mathbb{R}^p : \exists C \in [\mathbf{C}] : C \otimes \mathbf{x} = \mathbf{z}(k)\}. \tag{6.7}$$

Remark 6.3 *Note that $\tilde{X}_{k|k}$ is the set of all states that may lead to $\boldsymbol{z}_k$ via the measurement model in one event step.*

In the second sub-stage, the conditional reach set $X_{k|k}$ is obtained by intersecting the sets $X_{k|k-1}$ and $\tilde{X}_{k|k}$, thus:

$$X_{k|k} = X_{k|k-1} \cap \tilde{X}_{k|k}. \tag{6.8}$$

This intersection can be calculated by computing the canonical form representation of the intersection of each DBM representing $X_{k|k-1}$ with each DBM representing $\tilde{X}_{k|k}$.

Remark 6.4 *In the second stage the new information $\boldsymbol{z}_k$ is used to update the set $X_{k|k-1}$. This can be associated to the update stage of the Bayesian approach.*

If the set $X_{k-1|k-1}$ can be represented by union of $q_{k-1|k-1}$ DBM, then $X_{k|k-1}$ can be represented by a union of $q_{k|k-1}$ DBM. The inverse image of a point $\mathbf{z}_k$ can always be represented by a union of $\tilde{q}_{k|k}$ DBM (see section 5.3.2). Moreover, the intersection of two sets represented by the union of finitely many DBM is again a union of finitely many DBM. Therefore, $X_{k|k}$ can be represented by a union of $q_{k|k}$ DBM. Therefore, it can be proved that if X_0 can be represented by a union of finitely many DBM, then $X_{k|k}$ can also be represented by a union of $q_{k|k}$ DBM for each $k \in \mathbb{N}$.

The complexity of each stage is given in the following. The worst-case complexity to compute $X_{k|k-1}$ is $\mathcal{O}(q_{k-1|k-1}n^{n+3})$ for autonomous systems and $\mathcal{O}(\bar{q}_{k-1|k-1}(n+m)^{n+3})$ for nonautonomous systems (see section 5.1). The worst-case complexity to compute $\tilde{X}_{k|k}$ is $\mathcal{O}(l(l+n)^{l+1})$ (see section 5.3.2). Given $X_{k|k-1}$ and $\tilde{X}_{k|k}$, assumed to be represented by a union of $q_{k|k-1}$ and $\tilde{q}_{k|k}$ DBM, respectively, the worst-case complexity to compute $X_{k|k}$ via equation (6.8) is $\mathcal{O}(q_{k|k-1}\tilde{q}_{k|k}n^3)$.

Example 6.5 *In this example the conditional reach sets of an uMPL system is computed. The system considered is described by:*

$$\begin{aligned} \boldsymbol{x}(k) &= A(k) \otimes \boldsymbol{x}(k-1), \\ \boldsymbol{z}(k) &= C(k) \otimes \boldsymbol{x}(k). \end{aligned}$$

Where,

$$A(k) \in \begin{pmatrix} [1,\ 3] & [3,\ 4] \\ [2,\ 3] & [2,\ 4] \end{pmatrix} \textit{ and } C(k) \in \begin{pmatrix} [1,\ 3] & [1.5,\ 2.5] \\ 1 & [1,\ 3] \end{pmatrix}.$$

The simulated[1] *state and measurement sequences are given in Table 3. Using the measurement sequence and the set of initial positions* $X_0 = \{\boldsymbol{x} \in \mathbb{R}^2 : 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1,\}$*, the conditional reach set* $X_{1|1}$ *is computed in the following.*

Table 3 – Simulated state and measurement sequences.

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
$\mathbf{x}(k)$	$\begin{pmatrix} 0.661 \\ 0.019 \end{pmatrix}$	$\begin{pmatrix} 3.783 \\ 3.635 \end{pmatrix}$	$\begin{pmatrix} 7.121 \\ 6.999 \end{pmatrix}$	$\begin{pmatrix} 10.160 \\ 9.791 \end{pmatrix}$	$\begin{pmatrix} 13.146 \\ 13.362 \end{pmatrix}$
$\mathbf{z}(k)$	–	$\begin{pmatrix} 6.148 \\ 6.349 \end{pmatrix}$	$\begin{pmatrix} 9.530 \\ 8.555 \end{pmatrix}$	$\begin{pmatrix} 13.001 \\ 11.160 \end{pmatrix}$	$\begin{pmatrix} 15.351 \\ 14.629 \end{pmatrix}$

First, note that the set of initial positions X_0 *and the measurement* $\boldsymbol{z}(1)$ *can be represented by the following DBM:*

$$D^{(X_0)} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & e_{\mathcal{B}} & e_{\mathcal{B}} \\ (1,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (1,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array}$$

$$D^{(Z_1)} = \begin{array}{c} \begin{array}{ccc} x_0 & z_1 & z_2 \end{array} \\ \begin{pmatrix} e_{\mathcal{B}} & (-6.148,\leq) & (-6.349,\leq) \\ (6.148,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (6.349,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{pmatrix} \end{array} \begin{array}{c} \\ x_0 \\ z_1 \\ z_2 \end{array}$$

[1] For the simulation, it was considered that the entries of the matrices $A(k)$ and $C(k)$ are uniformly distributed in the given intervals. For example, for each k, the element $a_{11}(k)$ is uniformly distributed between 1 and 3.

In the first stage we compute $X_{1|0} = \mathcal{I}_{[\boldsymbol{A}]}\{X_0\}$, which can be represented by the collection of DBM given by $\mathtt{D}^{(X_{1|0})} = \{D^{(X^1_{1|0})}\}$, where:

$$D^{(X^1_{1|0})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-3,\leq) & (-2,\leq) \\ (5,\leq) & e_{\mathcal{B}} & (2,\leq) \\ (5,\leq) & (1,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array}$$

In the second stage we compute $\tilde{X}_{1|1} = \mathcal{I}^{-1}_{[\boldsymbol{C}]}\{\boldsymbol{z}_1\}$, which can be represented by the collection of DBM given by $\mathtt{D}^{(\tilde{X}_{1|1})} = \{D^{(\tilde{X}^1_{1|1})},\ D^{(\tilde{X}^2_{1|1})}\}$, where:

$$D^{(\tilde{X}^1_{1|1})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-3.147,\leq) & (-3.349,\leq) \\ (5.147,\leq) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (4.647,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array}$$

$$D^{(\tilde{X}^2_{1|1})} = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} & (-3.647,\leq) \\ (3.147,<) & e_{\mathcal{B}} & \varepsilon_{\mathcal{B}} \\ (4.647,\leq) & \varepsilon_{\mathcal{B}} & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array}$$

Finally, we compute $X_{1|1} = X_{1|0} \cap \tilde{X}_{1|1}$. This can be done by computing the canonical form representation of the intersection of each DBM in $\mathtt{D}^{(X_{1|0})}$ with each DBM in $\mathtt{D}^{(\tilde{X}_{1|1})}$, which yields:

$$D^{(X^1_{1|1})} = cf(D^{(X^1_{1|0})} \oplus_{\mathcal{B}} D^{(\tilde{X}^1_{1|1})}) = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-3.147,\leq) & (-3.349,\leq) \\ (5,\leq) & e_{\mathcal{B}} & (1.651,\leq) \\ (4.647,\leq) & (1,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array}$$

$$D^{(X^2_{1|1})} = cf(D^{(X^1_{1|0})} \oplus_{\mathcal{B}} D^{(\tilde{X}^2_{1|1})}) = \begin{array}{c} \begin{array}{ccc} x_0 & x_1' & x_2' \end{array} \\ \left(\begin{array}{ccc} e_{\mathcal{B}} & (-3,\leq) & (-3.647,\leq) \\ (3.147,<) & e_{\mathcal{B}} & (-0.5,<) \\ (4.147,<) & (1,\leq) & e_{\mathcal{B}} \end{array}\right) \end{array} \begin{array}{c} x_0 \\ x_1' \\ x_2' \end{array}$$

The conditional reach sets $X_{k|k}$ for $k \in \{1,2,3,4,496,497,498,499\}$ are shown in Figure 15. Note that the conditional reach set $X_{29|29}$ can be represented by a single DBM which illustrates that the number of DBM does not necessarily increases with k.

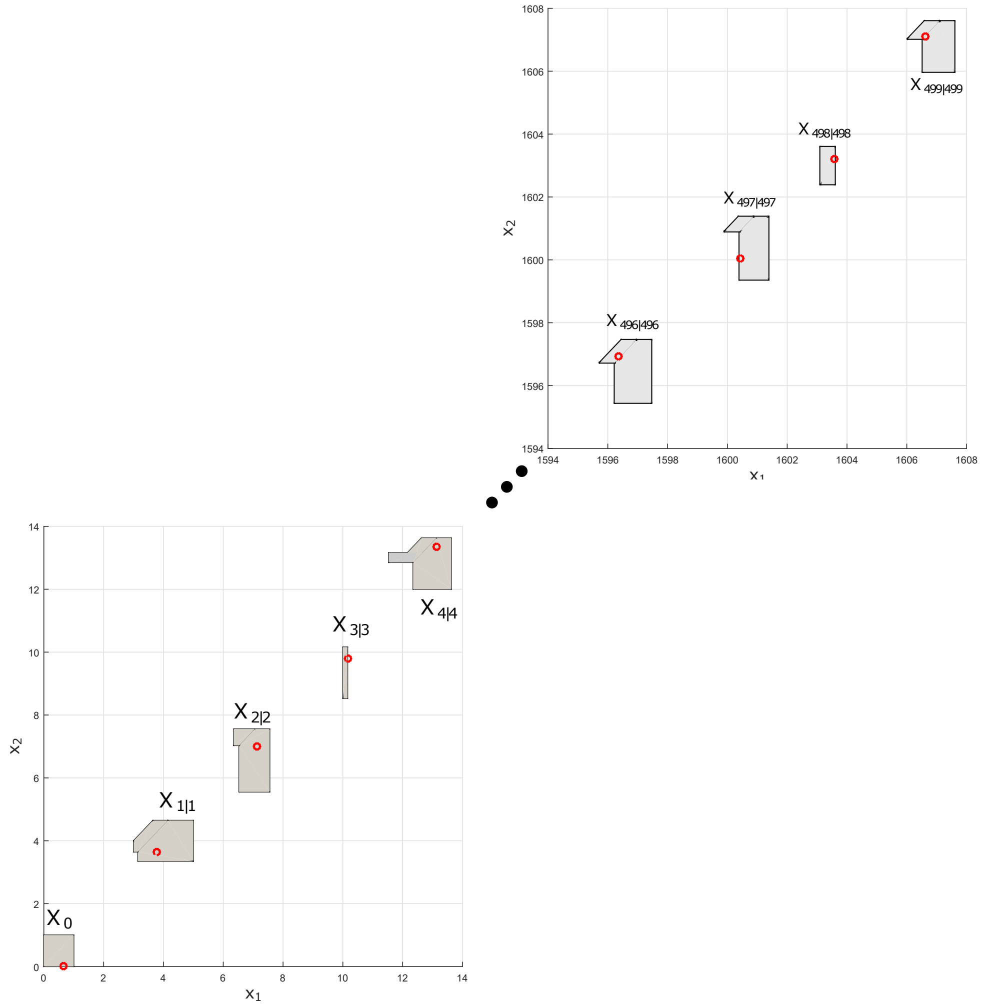

Figure 15 – conditional reach sets. The circles represent the real state values obtained via simulation.

As discussed in remark 5.8, the uMPL systems are expansive, i.e., the hyper-volume of the reach sets X_k tends to increase with k. However, the conditional reachability analysis uses the measurement model as a feedback mechanism which may avoid a potential explosion in the hyper-volume of the conditional reach sets. For the system considered in Example 6.5, for instance, it seems that the potential explosion will not happen (see Figure 15). However, as illustrated in Example 6.6, it is not the case for all systems.

Example 6.6 *Consider two uMPL systems characterized by the matrices presented in Table 4*

Considering $X_0 = \{\boldsymbol{x} \in \overline{\mathbb{R}}^2_{max} : 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1\}$, *the conditional reach sets* $X_{k|k}$, *for* $k \in \{1, \ldots, 59\}$, *were computed (for both systems). In order to observe if the conditional reach sets expand with* k, *they were plotted in Figure 16.*

Table 4 – Two uMPL systems.

	System 1	System 2
$[\mathbf{A}]$	$\begin{pmatrix} [3,\ 5] & \varepsilon \\ \varepsilon & [1,\ 11] \end{pmatrix}$	$\begin{pmatrix} [3,\ 5] & [0,\ 2] \\ \varepsilon & [1,\ 11] \end{pmatrix}$
$[\mathbf{C}]$	$\begin{pmatrix} [0,\ 2] & \varepsilon \end{pmatrix}$	$\begin{pmatrix} [0,\ 2] & \varepsilon \end{pmatrix}$

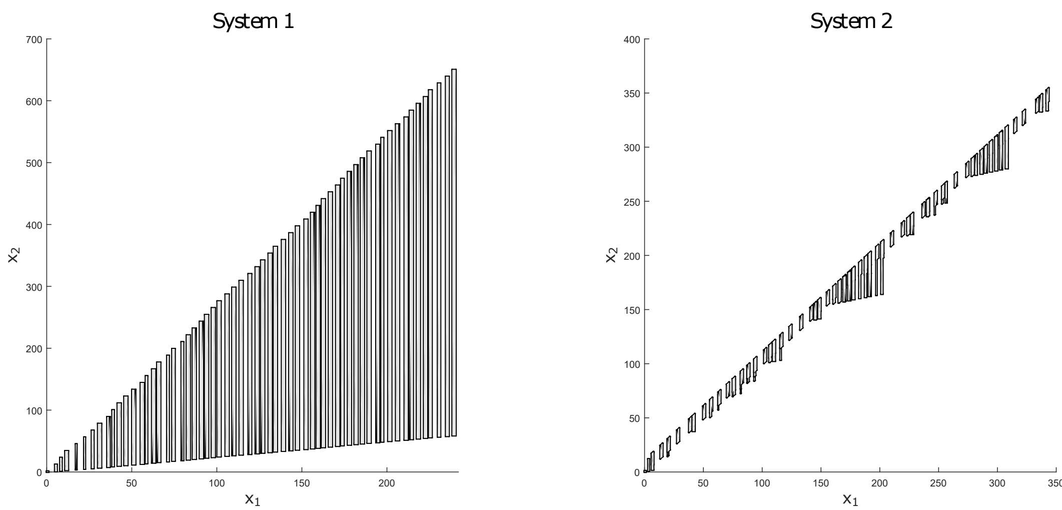

Figure 16 – Conditional reach sets for $k \in \{1, ..., 59\}$.

Note that, the conditional reach sets corresponding to System 1 clearly expand with k while System 2 seems to be nonexpansive. However, a question remains to be answered: under which conditions the system will be guaranteed nonexpansive? A sufficient condition is that the transition matrix $[\mathbf{A}] = [\underline{A},\ \overline{A}]$ be cyclic, i.e., the matrices of lower and upper bounds, $\underline{A}$ and $\overline{A}$, respectively, must to be irreducible matrices with the same cyclicity and max-plus eigenvalue (see section 5.1.1). However, it may not be a necessary condition as can be observed in Example 6.5, where the matrix $[\mathbf{A}]$ is not cyclic and the system seems to be nonexpansive.

7 Conclusion

Reachability analysis of MPL systems can be assessed by characterizing the system as PWA systems, which can be fully represented by DBM. DBM provide a simple and computationally advantageous representation of the MPL dynamics. Furthermore DBM are useful in reachability analysis of MPL systems since they can used to represent reach and backward reach sets. The main contribution of this thesis is to present a procedure to partition the state space of an uMPL system into components that can be completely represented by DBM. This has lead us to be able to present a procedure for computing the image and the inverse image of a DBM w.r.t. each component of the partitioned uMPL system which is similar to the procedure of computing the image and the inverse image of a DBM w.r.t. each component of a PWA system generated by a MPL system. Consequently, most of the previous results on reachability analysis of MPL systems could be extended to uMPL systems. The algorithms proposed have the same worst-case complexity as the algorithms proposed in (ADZKIYA *et al.*, 2014b; ADZKIYA *et al.*, 2014a; ADZKIYA *et al.*, 2015), with the advantage of handling a broader class of MPL systems. We shall note that, although the DBM-approach may be computationally expensive, it yields the exact reach sets. Therefore, it can be used as a benchmark to more conservatives and less expensive approaches.

In Chapter 6, we have presented an application of reachability analysis of uMPL systems. The forward and backward reachability analysis were used to solve the conditional reachability problem. Closely related to conditional reachability is the filtering problem, where one aims to construct the the posterior Probability Density Function (PDF) of the states based on all information available. The conditional reachability analysis corresponds to the support calculation of the posterior PDF.

As future work we aim to use the conditional reachability analysis to develop efficient filtering procedures for uMPL systems. Moreover, it seems viable the design of state-feedback controllers for uMPL systems, based on the knowledge of the support of the posterior PDF of the uMPL systems states.

Bibliography

ADZKIYA, D.; De Schutter, B.; ABATE, A. Computational techniques for reachability analysis of max-plus-linear systems. *Automatica*, v. 53, n. 3, p. 293–302, 2015. 16, 17, 34, 36, 40, 41, 48, 49, 50, 51, 52, 54, 55, 77, 117

ADZKIYA, D.; SCHUTTER, B. De; ABATE, A. Backward reachability of autonomous max-plus-linear systems. In: *Proceedings of the 12th IFAC/IEEE Workshop on Discrete Event Systems*. Cachan, France: [s.n.], 2014. p. 117–122. 16, 17, 48, 54, 55, 77, 117

ADZKIYA, D.; SCHUTTER, B. De; ABATE, A. Forward reachability computation for autonomous max-plus-linear systems. In: *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2014)*. Grenoble, France: [s.n.], 2014. p. 248–262. 15, 16, 17, 48, 51, 52, 53, 77, 117

ADZKIYA, D.; ZHANG, Y.; ABATE, A. Verisimpl 2: An open-source software for the verification of max-plus-linear systems. *Discrete Events Dynamical Systems*, 2015. 42

ALLAMIGEON, X.; GAUBERT, S.; GOUBAULT, E. Inferring min and max invariants using max-plus polyhedra. In: *Proceedings of the 15th International Symposium on Static Analysis*. Berlin, Heidelberg: Springer-Verlag, 2008. (SAS'08), p. 189–204. ISBN 978-3-540-69163-1. 15

AMARI, S.; DEMONGODIN, I.; LOISEAU, J. J.; MARTINEZ, C. Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, v. 57, n. 2, p. 462–467, Feb 2012. ISSN 0018-9286. 14

ARULAMPALAM, M.; MASKELL, S.; GORDON, N.; CLAPP, T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, v. 50, n. 2, p. 174 –188, feb 2002. ISSN 1053-587X. 17, 111

BACCELLI, F.; COHEN, G.; OLSDER, G.; QUADRAT, J. *Synchronization and Linearity : An Algebra for Discrete Event Systems*. [S.l.]: Wiley and Sons, 1992. 14, 18, 19, 20, 21, 23, 24, 25, 26, 27

BEMPORAD, A.; FERRARI-TRECATE, G.; MORARI, M. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, v. 45, n. 10, p. 1864–1876, Oct 2000. ISSN 0018-9286. 36

BRUNSCH, T.; HARDOUIN, L.; MAIA, C. A.; RAISCH, J. Duality and interval analysis over idempotent semirings. *Linear Algebra and its Applications*, v. 437, p. 2436–2454, 2012. ISSN 0024-3795. 63, 64

CÂNDIDO, R. M. F.; HARDOUIN, L.; LHOMMEAU, M.; MENDES, R. S. Conditional reachability of uncertain max plus linear systems. *Automatica*, 2017, Under Review to Automatica. 17

CÂNDIDO, R. M. F.; MENDES, R. S. Filtro de partículas para sistemas max plus com densidade de importância Ótima. In: *Anais do XX Congresso Brasileiro de Automática*. [S.l.: s.n.], 2014. p. 3413–3420. 17, 111

CÂNDIDO, R. M. F.; MENDES, R. S.; HARDOUIN, L.; MAIA, C. Particle filter for max-plus systems. *European Control Conference, ECC 2013*, Zurich, 2013. 17, 111

CASSANDRAS, C.; LAFORTUNE, S. *Introduction to Discrete Event Systems*. [S.l.]: Springer US, 2009. (SpringerLink Engineering). ISBN 9780387333328. 14

CASSANDRAS, C. G.; LAFORTUNE, S.; OLSDER, G. J. Introduction to the modelling, control and optimization of discrete event systems. In: ______. *Trends in Control: A European Perspective*. London: Springer London, 1995. p. 217–291. ISBN 978-1-4471-3061-1. 24

CHUA, L. O.; DENG, A. C. Canonical piecewise-linear representation. *IEEE Transactions on Circuits and Systems*, v. 35, n. 1, p. 101–111, Jan 1988. ISSN 0098-4094. 36

COHEN, G.; GAUBERT, S.; QUADRAT, J.-P. Max-plus algebra and system theory: Where we are and where to go now. *Annual Reviews in Control*, v. 23, p. 207 – 219, 1999. ISSN 1367-5788. 14

COHEN, G.; MOLLER, P.; QUADRAT, J.-P.; VIOT, M. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, v. 77, n. 1, p. 39–85, 1989. ISSN 0018-9219. 18, 20, 21, 22, 23

DILL, D. Timing assumptions and verification of finite-state concurrent systems. In: SIFAKIS, J. (Ed.). *Automatic Verification Methods for Finite State Systems*. [S.l.]: Springer Berlin Heidelberg, 1990, (Lecture Notes in Computer Science, v. 407). p. 197–212. ISBN 978-3-540-52148-8. 15, 29, 30, 31, 32, 34

DILORETO, M.; GAUBERT, S.; KATZ, R.; LOISEAU, J.-J. Duality between invariant spaces for max-plus linear discrete event systems. *SIAM Journal on Control and Optimization*, Society for Industrial and Applied Mathematics, v. 48, n. 8, p. 5606–5628, dez. 2010. 15, 62

DOUCET, A.; GODSILL, S.; ANDRIEU, C. On sequential monte carlo sampling methods for bayesain filtering. *Statistics and Computing*, Statistics and Computing, v. 10, 2000. 17, 111

DUBREIL, P.; DUBREIL-JACOTIN, M. L. *Leçons d'Algèbre Moderne*. 2. ed. Paris,France: Dunod, 1964. 20

FLOYD, R. W. Algorithm 97: Shortest path. *Communications of the ACM*, ACM Press, v. 5, n. 6, p. 345, 1962. 32

GAUBERT, S.; KATZ, R. Reachability and invariance problems in max-plus algebra. In: *Positive Systems, Proceedings of the First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA 2003), Rome, Italy*. [S.l.: s.n.], 2003. p. 15–22. 15

GAUBERT, S.; PLUS, M. Methods and applications of (max,+) linear algebra. In: _____. *STACS 97: 14th Annual Symposium on Theoretical Aspects of Computer Science Lübeck, Germany February 27–March 1, 1997 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. p. 261–282. ISBN 978-3-540-68342-1. 27

GAZARIK; MICHAEL, J.; KAMEN; EDWARD, W. Reachability and observability of linear systems over max-plus. *Kybernetika*, Institute of Information Theory and Automation AS CR, v. 35, n. 1, p. [2]–12, 1999. 15

GNING, A.; MIHAYLOVA, L.; ABDALLAH, F.; RISTIC, B. Particle filtering combined with interval methods for tracking applications. In: MALLICK, M.; KRISHNAMURTHY, V.; VO, B.-N. (Ed.). *Integrated Tracking, Classification, and Sensor Management*. [S.l.]: John Wiley and Sons, 2012. p. 43–74. 63

GONÇALVES, V. M.; MAIA, C. A.; HARDOUIN, L. On the steady-state control of timed event graphs with firing date constraints. *IEEE Transactions on Automatic Control*, v. 61, n. 8, p. 2187–2202, Aug 2016. ISSN 0018-9286. 14

GORDON, N.; SALMOND, D.; SMITH, A. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, v. 140, n. 2, p. 107 –113, apr 1993. ISSN 0956-375X. 17, 110

HARDOUIN, L.; COTTENCEAU, B.; LHOMMEAU, M.; CORRONC, E. L. Interval systems over idempotent semiring. *Linear Algebra and its Applications*, v. 431, n. 5-7, p. 855–862, ago. 2009. <doi:10.1016/j.LAA.2009.03.039>. 63, 64

HARDOUIN, L.; MAIA, C. A.; COTTENCEAU, B.; LHOMMEAU, M. Max-plus Linear Observer: Application to manufacturing Systems. In: *, WODES'10*. [S.l.: s.n.], 2010. p. 171–176. <istia.univ-angers.fr/~hardouin/Observer.html>. 15, 62

HEEMELS, W. P. M. H.; De Schutter, B.; BEMPORAD, A. Equivalence of hybrid dynamical models. *AUTOMATICA*, v. 37, p. 1085–1091, 2001. 16, 36

HEIDERGOTT, B.; OLSDER, G.; WOUDE, J. van der. *Max Plus at Work: Modeling and Analysis of Synchronized Systems : a Course on Max-Plus Algebra and Its Applications*. [S.l.]: Princeton University Press, 2006. (Max Plus at work: modeling and analysis of synchronized systems : a course on Max-Plus algebra and its applications, v. 13). ISBN 9780691117638. 14, 27, 28, 53

HEIDERGOTT, B. F. *Max-Plus Linear Stochastic Systems and Perturbation Analysis (The International Series on Discrete Event Dynamic Systems)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387352066. 15, 62

JOHANSSON, M.; RANTZER, A. Computation of piecewise quadratic lyapunov functions for hybrid systems. In: *Control Conference (ECC), 1997 European*. [S.l.: s.n.], 1997. p. 2005–2010. 36

JULIAN, P. The complete canonical piecewise-linear representation: functional form for minimal degenerate intersections. *IEEE Transactions on Circuits and Systems I:*

Fundamental Theory and Applications, v. 50, n. 3, p. 387–396, Mar 2003. ISSN 1057-7122. 36

KATZ, R. D. Max-plus (a,b)-invariant spaces and control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, v. 52, n. 2, p. 229–241, Feb 2007. ISSN 0018-9286. 14

KEVENAAR, T. A. M.; LEENAERTS, D. M. W. A comparison of piecewise-linear model descriptions. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, v. 39, n. 12, p. 996–1004, Dec 1992. ISSN 1057-7122. 36

LHOMMEAU, M.; HARDOUIN, L.; COTTENCEAU, B. Optimal control for (max, +)-linear systems in the presence of disturbances. In: *Positive Systems, Proceedings of the First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA 2003), Rome, Italy, August 28-30, 2003*. [S.l.: s.n.], 2003. p. 47–54. 14

LHOMMEAU, M.; HARDOUIN, L.; FERRIER, J.-L.; OUERGHI, I. Interval Analysis in Dioid : Application to Robust Open Loop Control for Timed Event Graphs. In: *and*. seville: [s.n.], 2005. p. 7744–7749. <doi:10.1109/CDC.2005.1583413>. 14, 63, 64

LITVINOV, G. L.; SOBOLEVSKIĪ, A. N. Idempotent interval analysis and optimization problems. *Reliable Computing*, v. 7, n. 5, p. 353–377, 2001. ISSN 1573-1340. 24, 63

LU, Q.; MADSEN, M.; MILATA, M.; RAVN, S.; FAHRENBERG, U.; LARSEN, K. G. Reachability analysis for timed automata using max-plus algebra. *The Journal of Logic and Algebraic Programming*, v. 81, n. 3, p. 298 – 313, 2012. ISSN 1567-8326. 15

MAIA, C. A.; HARDOUIN, L.; SANTOS-MENDES, R.; LOISEAU, J. J. A super-eigenvector approach to control constrained max-plus linear systems. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. [S.l.: s.n.], 2011. p. 1136–1141. ISSN 0191-2216. 14

MITCHELL, I. M. Comparing forward and backward reachability as tools for safety analysis. In: ______. *Hybrid Systems: Computation and Control: 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 428–443. ISBN 978-3-540-71493-4. Disponível em: <http://dx.doi.org/10.1007/978-3-540-71493-4_34>. 15

MOORE, R. E.; BIERBAUM, F. *Methods and Applications of Interval Analysis (SIAM Studies in Applied and Numerical Mathematics) (Siam Studies in Applied Mathematics, 2.)*. [S.l.]: Soc for Industrial & Applied Math, 1979. ISBN 0898711614. 63, 64

NECOARA, I.; De Schutter, B.; van den Boom, T.; HELLENDOORN, H. Robust control of constrained max-plus-linear systems. *Int. J. of Robust and Nonlinear Control*, v. 19, n. 2, p. 218–242, jan. 2009. 14

OLSDER, G. J.; RESING, J. A. C.; VRIES, R. E. D.; KEANE, M. S.; HOOGHIEMSTRA, G. Discrete event systems with stochastic processing times. *IEEE Transactions on Automatic Control*, v. 35, n. 3, p. 299–302, Mar 1990. ISSN 0018-9286. 15, 62

QUADRAT, J. *Semi-anillos en matematica aplicada*. [S.l.]: Universidad Nacional de Rosario, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, 1999. v. 28. (Cuadernos del Instituto de Matemática "Beppo Levi", v. 28). 19

RESING, J.; VRIES, R. de; HOOGHIEMSTRA, G.; KEANE, M.; OLSDER, G. Asymptotic behavior of random discrete event systems. *Stochastic Processes and their Applications*, v. 36, n. 2, p. 195 – 216, 1990. ISSN 0304-4149. 62

RIDI, L.; TORRINI, J.; VICARIO, E. Developing a scheduler with difference-bound matrices and the floyd-warshall algorithm. *IEEE Software*, v. 29, n. 1, p. 76–83, 2012. 31

SILVA, D. F.; MENDES, R. S.; HARDOUIN, L.; MAIA, C. A.; COTTENCEAU, B. Filtragem estocástica aplicada a sistemas max-plus lineares. In: *Anais do X Simpósio Brasileiro de Automação Inteligente*. [S.l.: s.n.], 2011. p. 1388–1393. 17, 111

SONTAG, E. D. Nonlinear regulation: The piecewise linear approach. *Automatic Control, IEEE Transactions on*, v. 26, n. 2, p. 346–358, Apr 1981. ISSN 0018-9286. 36

van den Boom, T.; De Schutter, B. Model predictive control for perturbed max-plus-linear systems. *Systems & Control Letters*, v. 45, n. 1, p. 21–33, jan. 2002. 15, 62

VANDENBERGHE, L.; MOOR, B. de; VANDEWALLE, J. The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits. In: *Circuits and Systems, 1989., IEEE International Symposium on*. [S.l.: s.n.], 1989. p. 2155–2158 vol.3. 36

WEN, C.; MA, X. A canonical piecewise-linear representation theorem: Geometrical structures determine representation capability. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 58, n. 12, p. 936–940, Dec 2011. ISSN 1549-7747. 36